人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-01

シジュウカラの言語能力と動物言語学の挑戦

Animal linguistics: Elements of language in the vocal communication of Japanese tits

鈴木俊貴

Toshitaka Suzuki

京都大学白眉センター・特定助教
京都大学創発 PI
The Hakubi Center for Advanced Research, Kyoto University, JST FOREST

概要

アリストテレスの時代から、「言語」はヒトと動物を隔てる決定的な性質であると考えられてきました。私たち人間は、単語を用いて物 事を示したり、それらを組み合わせて文章をつくり会話しますが、動物の鳴き声は単なる感情の表れにすぎないと捉えられてきたのです。 しかし、この二分は本当に正しいのでしょうか？　たしかに、感情を伝える音声だけでもうまく意思疎通がとれる動物もいるでしょうが、研究者たちは動物たちのコミュニケーションをどれほど詳細に解明できているのでしょうか？　私は、この疑問を胸に、シジュウカラの鳴き声について研究を続けてきました。

シジュウカラは都市部から山地まで広く見られる私たちに身近な野鳥です。15 年以上にわたる野外研究の成果として、シジュウカラは 捕食者の種類を示したり仲間を集めたりするための様々な鳴き声をもつことがわかってきました。さらに、彼らはこれらの鳴き声を組み合わせ、より複雑なメッセージ作ることまでできるのです。

さらに、野外において認知実験をおこなうことで、聞き手のシジュウカラは、鳴き声の示す対象をイメージしたり、音列に文法のルール を当てはめることで情報を読み解いていることもわかってきました。これらの発見は、私たちが普段会話のなかで使っている認知能力を動物において初めて実証した成果であり、言語の進化に迫る上でも大きな糸口を与えるはずです。

本講演では、上記の研究内容を紹介しながら、野外観察や行動実験から動物たちの豊かな会話の世界にどのように迫れるのか、その新たな学問の枠組み（動物言語学）についてもご紹介したいと思います。

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-02

A soundscape analysis of bird and cicada vocalizations based on azimuth and elevation localization using robot audition techniques

Hao Zhao[1*] Reiji Suzuki[1] Takaya Arita[1] Kazuhiro Nakadai[2]

[1] Nagoya University
[2] Tokyo Institute of Technology

Abstract: This study aims to apply robot audition techniques to investigate the natural soundscape of forest animal vocalizations based on azimuth-elevation estimation of sounds. We focus on two recordings in an experimental forest in Japan, where a 16-channel semi-spherical microphone array had been placed, and birds and cicadas dominated the soundscape. We manually annotated the localization results obtained from sound source localization and separation based on HARK, using HARKBird. Then we conducted a preliminary analysis on their vocal activity and the azimuth and elevation information. The result showed the difference in the soundscape patterns of their vocalizations in the azimuth-elevation space: a few birds tended to stay and sing at their song posts, and cicadas formed complex and bursty singing behavior, changing their positions. The behavior of cicada vocalizations may have reduced the vocal activity of birds that share the frequency ranges of their vocalizations while needs further detailed analyses to conclude.

Ecoacoustics [1] is an interdisciplinary field that investigates natural and human sounds and their relationships with an environment, which contributes to long-term ecosystem monitoring, habitat conservation, biodiversity assessment, and ecosystem management.

When considering the roles of sounds in ecoacoustics, the soundscape is an important concept that refers to the combination of sounds that arise from both natural and artificial environments [2]. Extracting a precise spatio-temporal structure of a soundscape and grasping the soundscape dynamics are essential for ecoacoustics to track active interactions among individuals and grasp the overall properties of the soundscape. However, it is not straightforward to grasp such a complex acoustic structure with a standard autonomous recording unit because it is hard to extract the spatio-temporal information of multiple sounds occurring in natural environments. Thus, there is increasing interest in microphone arrays to localize animal vocalizations [3, 4].

We have proposed and discussed novel applications of robot audition techniques to investigate the soundscape dynamics in the directional or spatial domain by using the direction of arrival (DOA) of sound sources obtained from HARKBird, which is a bird song localization software based on the robot audition software HARK (explained later) [5, 6]. We visualized and quantified the directional and 2D-spatial patterns of bird vocalizations in various contexts such as response behaviors in playback experiments [7, 8], for example.

As a next approach, we have been focusing on the soundscape dynamics of multiple classes of species dominating the soundscape of forests in early summer: birds and cicadas. It has been reported that birds are able to adjust both the timing and frequency of their signals to reduce overlap with the signals of other bird species [9, 10, 11], other animals[12] and abiotic noise [13]. Hart et al. showed that birds significantly avoid temporal overlap with cicadas by reducing and often shutting down vocalizations at the onset of cicada signals that utilize the same frequency range [12]. We used a method to classify their vocalizations using three ecoacoustic indices (acoustic complexity index, temporal entropy, and acoustic cover), then illustrated their temporal vocal activities, measured as the total song duration in each time segment [14]. As a proof-of-concept, we applied this to three scenarios of recordings to see if there exist inter-specific interactions between birds and cicadas and replayed the vocalizations of the cicadas to observe the effect on their vocalization activities. The preliminary analysis implied that there might exist temporal overlap avoidance behaviors between birds and cicadas, and replayed songs of cicadas may reduce the activity of birds.

While the above previous work is based on the azimuth estimation of acoustic events, there also exist variations in elevations among natural sound sources. Pekin et al. demonstrated the use of LIDAR-derived metrics and sound recordings for identifying canopy structural attributes supporting high acoustic diversity in a neotropical forest environment and showed that the composition of acoustic frequency bands and acoustic diversity are strongly linked with the vertical structure of the local forested environments [15]. Zezhou et al. used the soundscape mapping to explore the habitat selection of bird communities in the context of spatial-temporal structural changes and showed the urban forest vertical structure had a great ef-

*Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8601
E-mail:zhao.hao.y0@s.mail.nagoya-u.ac.jp

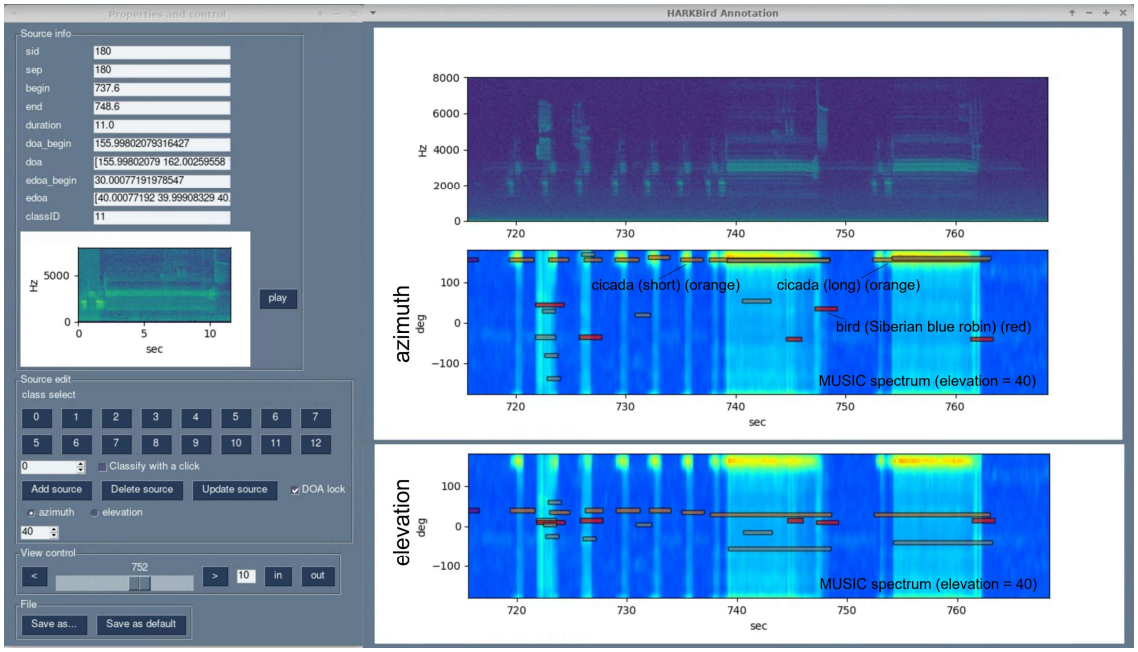


Figure 1: A snapshot of the annotation tool of HARKBird and examples of localized bird and cicada vocalizations.

fect on bird activities [16]. Yamamoto et al. visualized the spatio-temporal structure of soundscape composed of anthophony, biophony and geophony using the azimuth-elevation estimation of sound sources and their separation based on HARK, and their classification based on a recurrent convolutional neural network trained for bird vocalization classification [17]. They discussed that these three classes of sounds can be classified by their elevation information. These studies clarified the importance of elevation information for field observation of soundscape dynamics.

This study aims to further consider the application of robot audition techniques to the observation of the natural soundscape of forest animal vocalizations based on azimuth-elevation estimation of sounds. We focus on the recordings in an experimental forest, Nagoya University, Japan, on June 2021, where a 16-channel semi-spherical microphone array (Chirpy type-S; System in Frontier Inc.) had been placed, and birds and cicadas dominated the soundscape. This waterproof and standalone microphone array enables us to record and localize the out-door soundscape in azimuth and elevation angles. As a preliminary approach, we manually annotated the localization results of two recordings obtained from sound source localization and separation based on HARK, using HARKBird. Then, we conducted a preliminary analysis on their vocal activity and the azimuth and elevation information of their vocalizations. The analysis implied that there might exist temporal overlap avoidance behaviors between birds and cicadas, as expected from our previous results, constructing the distinctive spatial structure of vocalization dynamics in the azimuth-elevation space of the forest soundscape.

1 Materials and methods

1.1 HARKBird

HARK is an open-sourced robot audition software consisting of multiple modules for sound source localization, sound source separation, and automatic speech recognition of separated sounds that work on any robot with any microphone configuration [18]. See the website of HARK for detail[1].

We used HARKBird [5, 6], a collection of Python scripts for bird song localization, to estimate the DOA of sound sources in recordings, using sound source localization and separation functions in HARK. We adopted the current version (3.0b), based on PySimpleGUI, which enables us to visualize and annotate the sound source distribution in the azimuth-elevation space as shown in Fig. 1.

The employed sound source localization algorithm is based on the multiple signal classification (MUSIC) [19] using multiple spectrograms obtained by short-time Fourier transformation (STFT). The MUSIC method is a widely used high-resolution algorithm and is based on the eigenvalue decomposition of the correlation matrix of multiple signals from a microphone array. We adopted the standard eigenvalue decomposition (SEVD) MUSIC method implemented as one of the sound source localization methods in HARK. All localized sounds are separated to the sounds as wave files (16 bit, 16 kHz) using geometric high-order decorrelation-based source separation (GHDSS) method [20], which is also implemented

[1] https://hark.jp

in HARK. For more details on HARKBird[2], see [5, 6]. To optimize localization performance, we can adjust some parameters of HARKBird, such as the source tracking and the lower bound frequency for MUSIC, to reduce noise, etc.

Figure 2: An experimental field and a recording node (Chirpy type-S).

1.2 Field recording

On June 2021, we conducted the recording experiments to observe the vocalizations of cicada and bird individuals at our field site in the Inabu field, the experimental forest of Field Science Center, Graduate School of Bioagricultural Sciences, Nagoya University, in central Japan (Figure 2). The forest is mainly composed of conifer plantations (Japanese cedar, Japanese cypress, and red pine), with small patches of broadleaf trees (quercus, acer, carpinus, etc.).

During the recording, the common bird and cicada species were known to vocalize during early summer. The Siberian blue robin (*Larvivora cyane*) was the species that mainly dominated the soundscape. There were also a single species of cicadas (*Terpnosia nigricosta*). The vocalization of this species has a unique structure that is composed of repetitions of introductory short components like frog calls, and subsequent main song components like songs of poplar evening cicadas. These bird and cicada species share the frequency ranges of their vocalizations.

We used a 16-channel microphone array (Chirpy type-S; System in Frontier Inc.) system for out-door long-term recording. The system is composed of a hemispherical microphone unit connected to the control box. The water-proofed 16 microphone channels were placed in a spiral manner on the surface of the unit, which allows us to estimate both azimuth and elevation angles of the localized sounds, using HARKBird. It can also record the posture information of the unit as an additional channel in the wave file. The control box has functions of power supplies based on multiple USB batteries, scheduled recording, and the Wi-Fi-based server system that allows us to configure the schedule and conduct manual recordings via a client software for Windows PCs.

We conducted experimental recordings on consecutive days this month. In this paper, we picked two recording sessions for analysis as examples of the natural soundscape. The first 10-min recording started at 9:00 am, on June 26th. The other 20-min recording started at 9:20 am, on June 24th in which sometimes the experimenter's activity sounds were included.

1.3 Sound source localization, separation, and manual annotation

We used the HARKBird to export the information on localized sound sources (i.e., the beginning and end time, DOA (azimuth and elevation), and its separated sound file (wave file)). To extract the overall spatio-temporal structure of the soundscape, we adjusted the other parameters in HARKBird to localize these vocalizations that allow us to localize most sound sources around the microphone array.

After sound localization and separation, we manually annotated the vocalization events of cicadas and birds by conducting an auditory and visual inspection of the localization and separation results of sound sources. Among the separated sounds, we picked up sources that could be associated with visually recognized bird and cicada vocalizations in the spectrogram. We removed other sources that had no clear associations, which were expected to appear due to irregular noises or effects of other sounds that increased the overall power of MUSIC spectrum. We mainly used the visualized MUSIC spectrum for this purpose. However, there is a possibility that some artifacts were misclassified as cicada vocalizations in this preliminary analyses. Also, there exist a possibility of effects of experimenters' effects on their behaviors.

1.4 Analysis

We visualized the temporal dynamics of the localized vocalizations of birds and cicadas in the azimuth and elevation space, and a circular histogram of the azimuth-elevation angles of localized sounds. These graphs enable us to grasp the overall distribution of their vocalizations.

In order to observe the relationship of the activities between birds and cicadas, we quantified the temporal changes in the vocal activities of birds and cicadas to observe inter-specific interactions between birds and cicadas. Their activity in each 30-second (for the 10-min recording) or 50-second (for the 20-min recording) was calculated as the total duration of localized sounds in the segment.

[2]http://www.alife.cs.i.nagoya-u.ac.jp/~reiji/HARKBird/

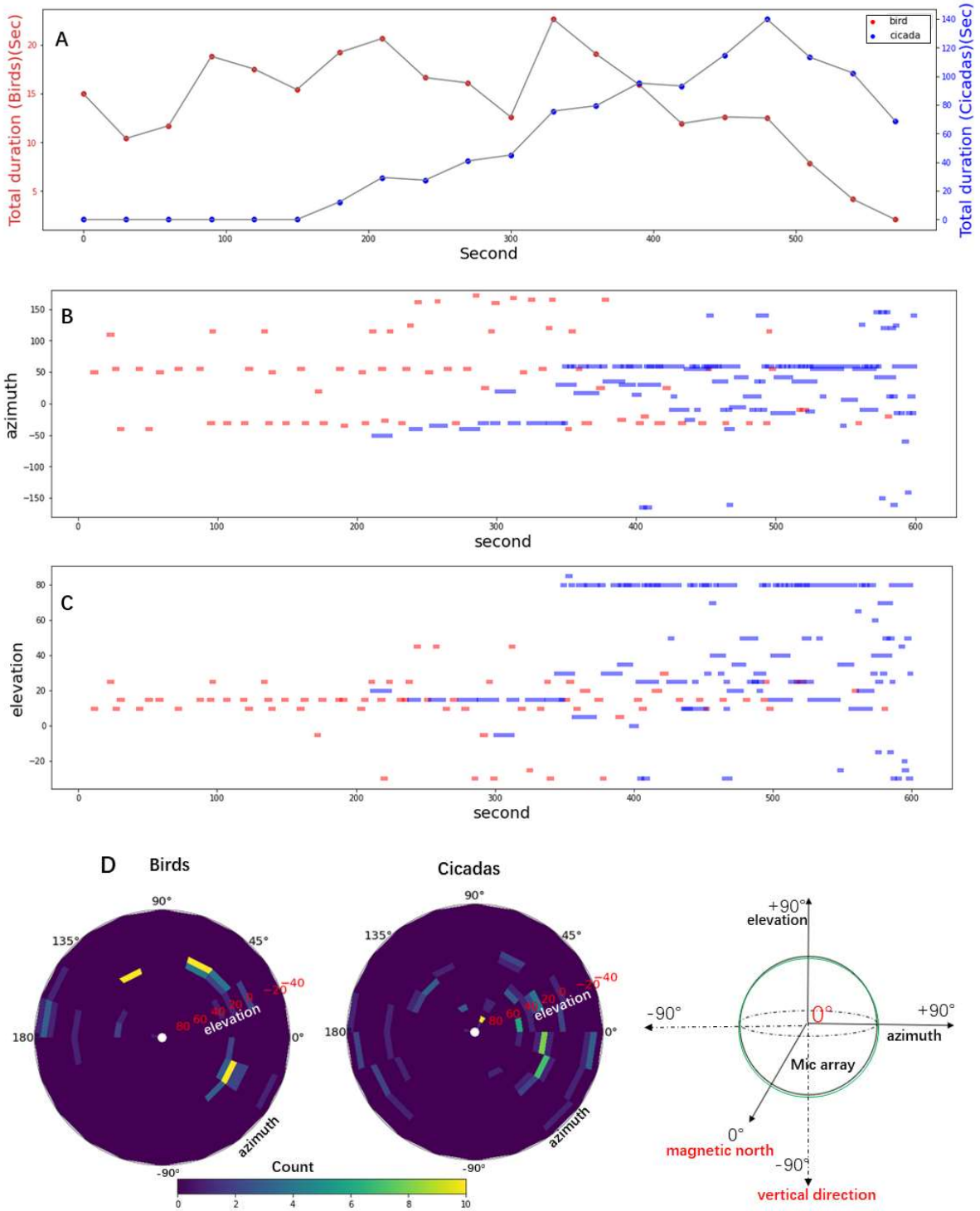


Figure 3: Vocal analysis for 10 minutes (9:00 am-9:10 am, June 26th).

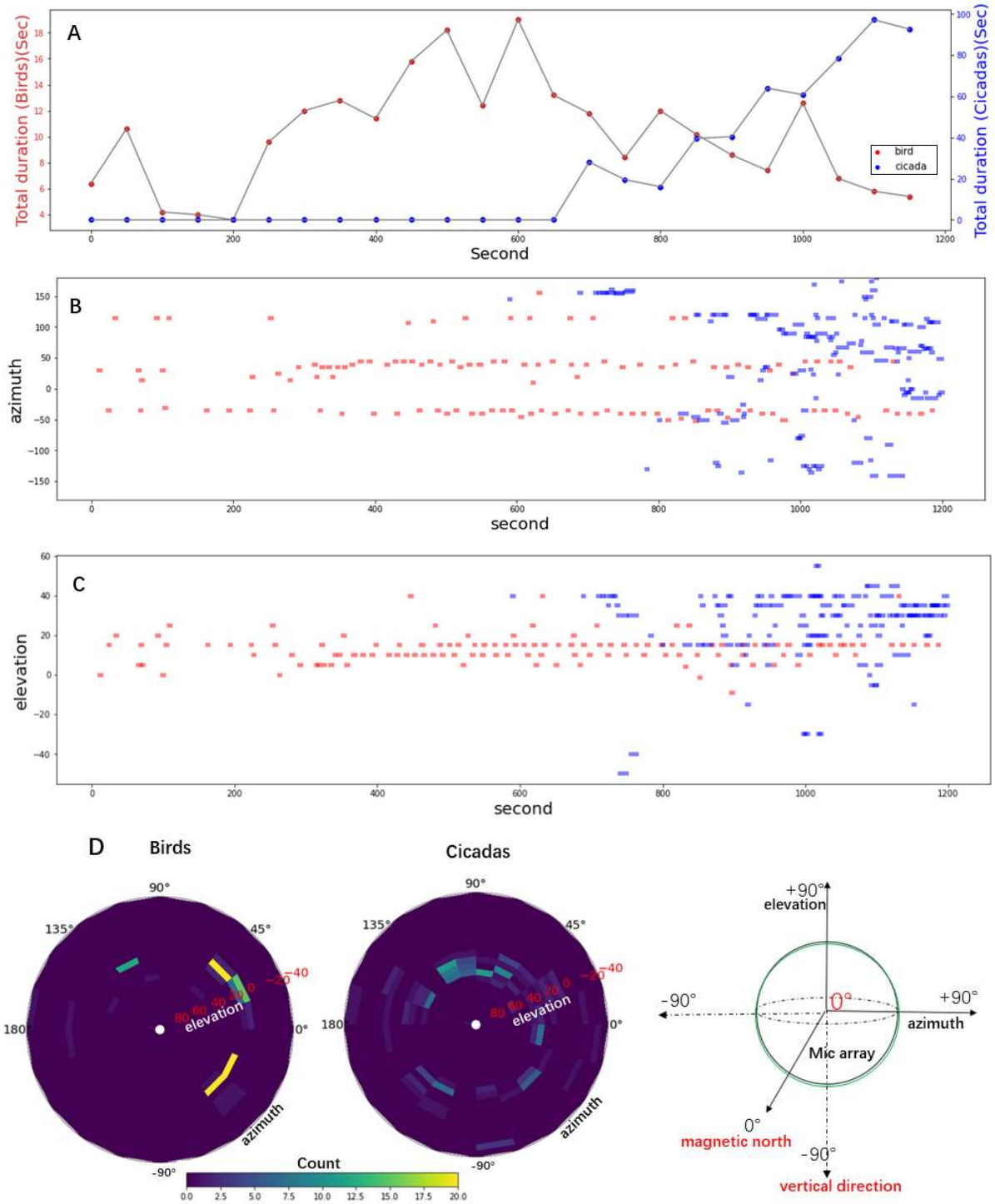


Figure 4: Vocal analysis for 20 minutes (9:20 am-9:40 am, June 24th).

2 Results

Fig. 3 and Fig. 4 show (A) the temporal changes in the vocal activity of the birds (red) and cicadas (blue), (B) the distribution of vocalizations in the space of time and azimuth, (C) the distribution of vocalizations in the space of time and elevation, and (D) the polar histograms of azimuth and elevation angles of bird and cicada vocalizations. All the bird songs were of Siberian blue robins in the two recordings.

In Fig. 3A and Fig. 4A, we can find the birds vocalized actively but the cicadas vocalized relatively quietly in the first half of the recordings. On the other hand, in the later half of both recordings, cicadas vocalized actively but the birds were gradually getting quiet. This indicates that the birds might tend to avoid vocalizing with cicadas.

In B and C of both Fig. 3 and 4, we see a few birds sang repeatedly at consistent positions (e.g., 0-200 seconds in Fig. 3 and 0-700 seconds in Fig. 4), which might be their song posts. Then, one cicada began to sing and the number of singing cicadas gradually increased, forming their chorus or burst. Their songs were composed of short and long components and their singing positions appeared to change both in azimuth and elevation. When their vocalization activity increased, some birds stopped singing or sang less frequently (e.g., after 400 seconds in Fig. 3 and after 900 seconds in Fig. 4), which made the activity of birds relatively small. This supports that these birds might have avoided singing with cicadas. However, further analyses based on long-term data are necessary to conclude.

Fig. 3D and 4D well illustrated the overall soundscape structures of bird and cicada vocalizations in the space of azimuth-elevation angles. We see that the birds at different azimuths vocalized at similar elevation angles. It reflects, in these particular cases, birds tended to sing in the middle of the trees because this is the typical behavior of Siberian blue robins. Another interesting fact is that their soundscape structures were similar between these two different days, which indicates that their habit use were consistent. On the contrary, we see that the soundscape structures of cicada vocalizations were more complex in the sense that there were large variations and frequent changes of localized positions in both azimuth and elevation angles. Some cicada sounds even came from places lower than the microphone because the microphone was placed around a small ridge. Also, the structures varied between both recordings, implying that their temporal dynamics were also complex. We expect that longer-term analyses will enable us to discuss the relationship between their soundscape structures.

3 Conclusion

This paper applied robot audition techniques to investigate the soundscape dynamics of cicada and bird vocalizations. We focus on two recordings and manually annotated the results of azimuth and elevation localization. Then we conducted a preliminary analysis on their vocal activity and the azimuth and elevation information of their vocalizations. The result showed the difference in the soundscape patterns of their vocalizations in the azimuth-elevation space: a few birds tended to stay and sing at their song posts, and cicadas formed complex and bursty singing behavior, changing their positions. The behavior of cicada vocalizations may have reduced the vocal activity of birds that share the frequency ranges of their vocalizations, which supports previous reports [12]. However, there exists a possibility of mislocalization of cicada vocalizations due to the interference of their loud and multiple vocalizations, which is the future work to be improved. The further analyses of their soundscape based on long-term data also are necessary.

Acknowledgements

We thank Naoki Takabe (Nagoya University) for supporting field experiments. This work was supported in part by JSPS/MEXT KAKENHI: JP21K12058, JP20H 00475, JP19KK0260.

References

[1] A. Farina and S. H. Gage. *Ecoacoustics: The Ecological Role of Sounds.* John Wiley and Sons, 2017.

[2] J.D. Douglas and J.F. Mastin. Soundscape. *Journal of Architectural and Planning Research*, pages 169–186, 1985.

[3] D. Blumstein, D. J. Mennill, P. Clemins, L. Girod, K. Yao, G. Patricelli, J. L. Deppe, A. H. Krakauer, C. Clark, K. A. Cortopassi, S. F. Hanser, B. McCowan, A. M. Ali, and A. N. G. Kirshel. Acoustic monitoring in terrestrial environments using microphone arrays: applications, technological considerations and prospectus. *Journal of Applied Ecology*, 48:758–767, 2011.

[4] T. A. Rhinehart, L. M. Chronister, T. Devlin, and J. Kitzes. Acoustic localization of terrestrial wildlife: Current practices and future opportunities. *Ecology and Evolution*, 10(13):6794–6818, 2020.

[5] R. Suzuki, S. Matsubayashi, K. Nakadai, and H. G. Okuno. HARKBird: Exploring acoustic interactions in bird communities using a microphone array. *Journal of Robotics and Mechatronics*, 27:213–223, 2017.

[6] S. Sumitani, R. Suzuki, S. Matsubayashi, K. Arita, T. Nakadai, and H. G. Okuno. An integrated framework for field recording, localization, classification and annotation of birdsongs using

robot audition techniques - HARKBird 2.0. In *Proceedings of ICASSP 2019*, pages 8246–8250, 2019.

[7] Reiji Suzuki, Shinji Sumitani, Naren , Shiho Matsubayashi, Takaya Arita, Kazuhiro Nakadai, and Hiroshi G. Okuno. Field observations of ecoacoustic dynamics of a japanese bush warbler using an open-source software for robot audition hark. *Journal of Ecoacoustics*, 2:EYAJ46, 2018.

[8] S. Sumitani, R. Suzuki, S. Matsubayashi, T. Arita, K. Nakadai, and H.G. Okuno. Fine-scale observations of spatio-spectro-temporal dynamics of bird vocalizations using robot audition techniques. *Remote Sensing in Ecology and Conservation*, rse2.152, 2020.

[9] M. L. Cody and J. H. Brown. Song asynchrony in neighbouring bird species. *Nature*, 222:778–780, 1969.

[10] H. Brumm. Signalling through acoustic windows: nightingales avoid interspecific competition by short-term adjustment of song timing. *Journal of Comparative Physiology A: Neuroethology*, 192:1279–1285, 2006.

[11] P. J. Hart, T. Ibanez, K. Paxton, G. Tredinnick, E. Sebastián-González, and A. Tanimoto-Johnson. Timing is everything: Acoustic niche partitioning in two tropical wet forest bird communities. *Frontiers in Ecology and Evolution*, 9:753363, 2020.

[12] P. J. Hart, R. Hall, W. Ray, A. Beck, and J. Zook. Cicadas impact bird communication in a noisy tropical rainforest. *Behavioral Ecology*, 26:839–842, 2015.

[13] H. Slabbekoorn and M. Peet. Birds sing at a higher pitch in urban noise. *Nature*, 424, 2003.

[14] H. Zhao, R. Suzuki, S. Sumitani, S. Matsubayashi, T. Arita, K. Nakadai, and H. G. Okuno. Visualizing soundscapes and quantifying interspecific interactions in forest animal vocalizations using robot audition technology. In *Proceedings of 84th Annual Meeting of IPSJ*, volume 2, pages 475–476, 2022.

[15] B. K. Pekin, J. Jung, L. J. Villanueva-Rivera, B. C. Pijanowski, and J.A. Ahumada. Modeling acoustic diversity using soundscape recordings and lidar-derived metrics of vertical forest structure in a neotropical rainforest. *Landscape ecology*, 27(10):1513–1522, 2012.

[16] Z. Hao, C. Wang, Z. Sun, C. K. Vandenbosch, D. Zhao, B. Sun, X. Xu, Q. Bian, Z. Bai, K. Wei, et al. Soundscape mapping for spatial-temporal estimate on bird activities in urban forests. *Urban Forestry & Urban Greening*, 57:126822, 2021.

[17] R. Yamamoto, K. Nishida, K. Itoyama, S. Matsubayashi, and K. Suzuki, R. Nakadai. Soundscape analysis using robot audition open source software hark. In *Proceedings of the Annual Meeting of Ornithological Society of Japan*, 2022.

[18] K. Nakadai, H. G. Okuno, and T. Mizumoto. Development, Deployment and Applications of Robot Audition Open Source Software HARK. *Journal of Robotics and Mechatronics*, 27:16–25, 2017.

[19] R. Schmidt. Bayesian nonparametrics for microphone array processing. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986.

[20] H. Nakajima, K. Nakadai, Y. Hasegawa, and H. Tsujino. Blind source separation with parameter-free adaptive step-size method for robot audition. *IEEE Transactions on Audio, Speech, and Language Processing*, 18:1476–1485, 2010.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-03

低解像度画像からの小領域物体の検出手法の検討

Investigation of a method for detecting small objects from low-resolution images

西田健次 [1*]　糸山克寿 [1,2]　中臺一博 [1]
Kenji Nishida[1], Katsutoshi Itoyama[1,2], Kazuhiro Nakadai[1]

[1] 東京工業大学
[1] Tokyo Institute of Technology
[2] ホンダ・リサーチ・インスティチュート・ジャパン
[2] Honda Research Institute Japan

Abstract: 魚眼レンズによる全天画像から鳥の行動を検出する手法を提案した．全天画像内での鳥の画像は小さく，変形しながら移動するため，物体追跡手法の適用は困難である．また，木の枝によるオクルージョンも鳥を検出することを難しくしている．本稿では，鳥自体の検出ではなく，鳥の行動に基づく画像上の動き（鳥自体の動きとともに，鳥の行動によって生じた木の枝の揺れ）のオプティカルフローを検出し，鳥の行動を検知することを目指している．鳥は，羽ばたきによる変形を伴い画像内を移動するため，鳥の移動方向とは異なるオプティカルフローを発生させている．そのため，単純にオプティカルフローを検出するだけでは鳥の行動を検知できない．また，風による木の枝の揺れと鳥の行動による木の枝の揺れを判別する必要がある．提案手法では，オプティカルフローの発生する領域の移動を検知することにより，風による木の揺れによって生じるオプティカルフローを排除し，鳥の行動検出を行っている．

1　はじめに

鳥の生態観測は，鳥類学での課題だけでなく，環境保護，航空機の安全確保など，様々な分野で必要とされている．そのため，鳥の鳴き声や映像を記録し，その鳥の三次元環境中での位置や種類を特定する様々な手法が開発されてきた [1, 2]．

画像情報からの鳥の位置推定は，風景画像内での鳥が小さくなってしまうため，検出および種類識別は簡単なものではなかったが，幾つかの提案がなされている．吉橋らは，CNN (Convolutional Neural Network) に基づく風力発電所での鳥の検出方法を提案しており [4]，また，複数の CNN 構造から得られる深層物体特徴を組み合わせることで風景シーン中で小さく見える鳥の検出に特化した手法も提案している [5]．鳴き声による位置推定も提案されてきており，Gayk らは，8 個のマイクから成るマイクロホンアレイによって，飛翔する都市の鳴き声の位置推定を行っている [3]．Vemeycken らは，64 個のマイクから成る，より高密度のマイクロホンアレイを用いて，半径 75 mの範囲での鳥の鳴き声の位置を定位することに成功した [6]．また，住谷らは，8 個のマイクから成るマイクロホンアレイ TAMAGO-03 を用い，３Ｄ環境での音声と映像の両者を収録し，鳴き声解析のためのソフトウェア HARKBird も開発している [7]．しかし，実環境で収録された音声データには，通常，ノイズが多く含まれるため，検出精度への影響が懸念される．川西らは，音声情報により大まかな位置推定を行い，その近傍の画像情報により鳥検出を行う，2 つのモダリティを組み合わせた手法を開発し，精度向上を目指している [8]．この手法では，ケージ内でのパノラマ画像内からアノテーションされた鳥の学習データセットを構築しており，鳥の姿が見える環境での有効性が示されている．

本稿で対象とする画像は，千葉県佐倉市内に設けた魚眼レンズによる半球画像であり，鳥の姿が小さいだけでなく，樹木による隠れが生じており，必ずしも鳥の姿を捉えることが可能ではない．そこで，画像内に鳥によって生じた動き（鳥自体の動き，あるいは，鳥による木の枝の動き）を捉えることにより，鳥の検出に代える手法を提案する．本提案手法では，鳥による木の枝の動きと，風による木の枝の動きを弁別し，鳥による動きのみを抽出することが要点となる．第 2 節にて本稿の対象とする画像データについてのべ，第 3 節では提案手法の詳細を述べる．第 4 節にて実験結果

*連絡先：東京工業大学
152-8552 東京都目黒区大岡山 2–12–1 W8-30
E-mail: nishida@sc.e.titech.ac.jp

図 1: 全天画像の例

図 2: 枝の動きによるオプティカルフロー

の概要を示し，第 5 節で結論と今後の課題について述べる．

2　画像データの概要

本稿で使用した画像データは，魚眼レンズによる全天（半球）画像であり，日出日没の前後 1 時間，その他 1 時間ごとに 10 分づつ録画したもので，同時に DACHO マイクロホンアレイでの多チャンネル録音も行っている．カメラは 4 か所に設置されており，木の多いもの，空の占める面積の大きいものなど，それぞれ異なる傾向の画像が収録されている（図 1）．

画像に現れる鳥の行動は，鳥の姿はあまり見えず木の枝を渡るもの，中空を飛ぶものに大別することができる．木の枝を渡るものは鳥の姿がほとんど見えないが，鳥の動きに伴い木の枝が動いている．(図 2)．一方，中空を飛ぶ鳥は，画像として小さく，変形しながら移動するため（図 3），物体検出アルゴリズムを適用することは簡単ではない．そこで，画像上の動き（オプティカルフロー）を手掛かりに，鳥の検出を行う手法を検討した．

図 3: 中空を飛ぶ鳥によるオプティカルフロー

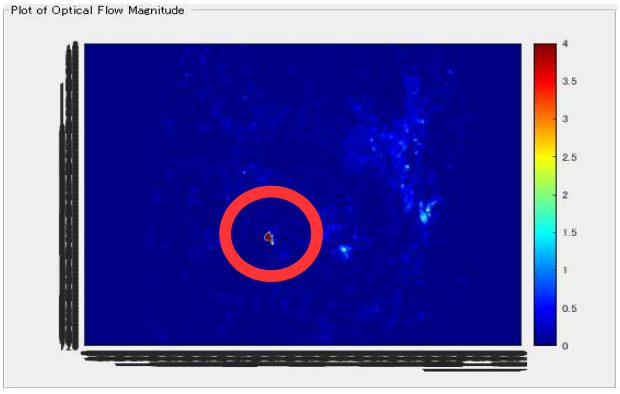


図 4: オプティカルフロー絶対値のヒートマップ

3 オプティカルフローによる鳥検出手法

本稿で対象とした画像は，検出対象となる対象が小さく，また，背景画像の部分に急峻な輝度勾配と細かいテクスチャがあるため，オプティカルフローの典型的な手法である Lucas-Kanade 法（[9]）では，充分なオプティカルフローを検出することができなかった．一方，Farneback 法（[10]）法でオプティカルフローを求めたが，細かい動きや変形をもとらえるため，飛んでいる鳥に対して鳥全体の動きではなく，羽ばたきなどの画像上での変形によるオプティカルフローを捉えてしまっている（図 3）．飛ぶ鳥によるオプティカルフローは鳥の飛行方向を示してはいないが，鳥の移動に伴いオプティカルフローを持つ領域が移動している．そこで，オプティカルフローを持つ領域の移動を検出することによって，鳥の検出を行う（図 4）．

オプティカルフローによる鳥検出を行う際に課題となるのは，風などによる木の動きによって生じるフローと鳥によって生じるフローの弁別を行う事である．鳥の行動に基づくオプティカルフローと風による木の枝の揺れのオプティカルフローの違いは以下のように考えられる．

- 鳥の活動に基づくオプティカルフローの発生する領域は，鳥の移動に伴い移動する
- 風による木の枝の揺れに基づくオプティカルフローは，ほぼ同じ位置で生じ，移動範囲は狭い

これにより，風によって生じるオプティカルフローは，その発生する領域が大きくは移動せず，長い時間帯では狭い範囲を移動していると考えられる．そこで，空間的（画像内）でのスムージングと時間方向でのスムージングを組み合わせることによって，フローを発生する領域の移動幅は小さく保つことができる．これにより，オプティカルフローを発生する領域が移動するものを，鳥の行動として検知できる．

提案手法の手順を以下に示す (図 [5]）．ただし，時刻 t のヴィデオフレームを $\boldsymbol{v_t}$ とし，$\mathcal{F}$ はオプティカルフローを求める関数（本稿では Farneback 法を用いた），$\mathcal{G}(\boldsymbol{x}, \sigma_s)$ は σ_s を分散とした二次元の平滑化関数，$G(s, \sigma_t)$ は偏差 s に対して σ_t を分散としたガウス関数，$\mathcal{B}(x, \delta)$ は δ を閾値とした二値化関数，$\mathcal{D}(\boldsymbol{x}, \boldsymbol{y})$ はベクトル $\boldsymbol{x}$ とベクトル $\boldsymbol{y}$ の要素間の距離を総当たりで出力する関数とする．

1. 時刻 t でのオプティカルフロー $\boldsymbol{f}_t = \mathcal{F}(\boldsymbol{v}_t)$ を求める
2. オプティカルフローの絶対値を分散 σ_s のガウス関数で平滑化する： $\boldsymbol{g}_t = \mathcal{G}(||\boldsymbol{f}_t||, \sigma_s)$
3. 空間的に平滑化されたオプティカルフロー $\boldsymbol{g}_t$ に対して，分散 σ_t のガウス重みで移動平均をとる：
$$\boldsymbol{h_t} = \sum_{i=t-s}^{t+s} G(s, \sigma_t)\boldsymbol{g}_i$$
4. 時空間的に平滑化されたオプティカルフローを二値化する: $\boldsymbol{b}_t = \mathcal{B}(\boldsymbol{h}_t, \delta)$
5. 二値画像中のオプティカルフローの存在する（複数の）領域の重心を求める： $\boldsymbol{c}_t = \mathcal{C}(\boldsymbol{b}_t)$
6. 5 で求めた領域重心と一つ前のフレームの領域重心との相互距離を測る: $\boldsymbol{D}_t = \mathcal{D}(\boldsymbol{c}_t, \boldsymbol{c}_{t-1})$
7. 時刻 t での領域重心 c_t^i から見て，時刻 $t-1$ の最も近い領域重心との距離を d_t^i として $\theta_l < d_t^i < \theta_h$ の時，c_t^i を，鳥によって生じたオプティカルフロー領域の重心とする

手順 4 の二値化により，オプティカルフローの存在する領域と存在しない領域の，画像上での分割が行われる．手順 5 から 7 は，手順 4 で求めたオプティカルフロー領域の重心の追跡を行っている．即ち，時刻 $t-1$ と時刻 t 間で最も近いフロー領域同士の距離を領域重心の移動量と考え，移動量が θ_l より小さな領域は枝の揺れによるフローであるとし，θ_h よりも移動量が大きなものは対応するフロー領域がなかったものとしている．

4 実験結果

風が強く樹木の動きが大きい場合の検出結果を図 6 に示す．オプティカルフロー検出では，風による木の動きによるフローが多数検出されているが，提案手法では鳥の移動によるフローの部分のみが検出できている．しかし，時間方向の平滑化を広く取り過ぎたため，鳥の移動速度に対応しきれていないシーケンスが存在した．

図 7 に，鳥の姿は枝に隠れている場合の，鳥による枝の動きの検出結果を示す．この例では，風による木の動きはほとんどないため，枝の動きによるフローを検出することで鳥の動き検出を行う．枝が単純に揺れている場合のフローは鳥の動きとしては検出されず，枝の揺れる領域（フローの生じる領域）が移動する場合に，鳥の動きとして検出できている．

5 結論と今後の課題

本稿では，低解像度で隠れが頻発する画像から，オプティカルフローを手掛かりに鳥の行動を検出する手

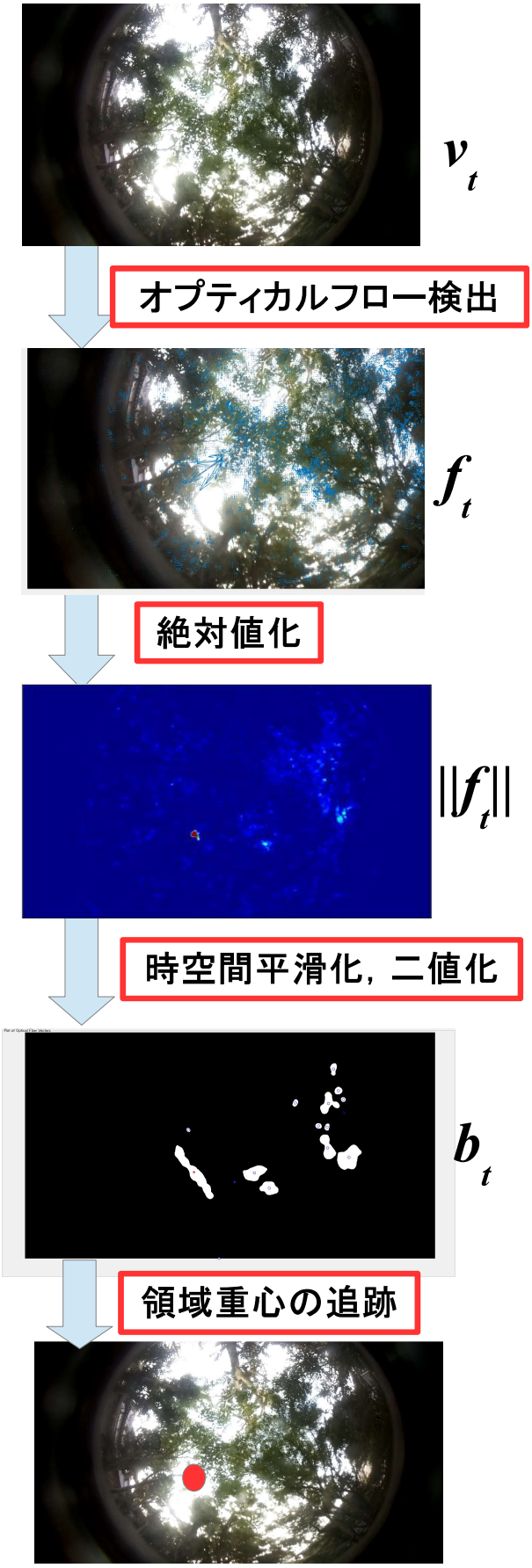


図 5: 鳥の動き検出の手順

図 6: 風の強い場合の検出

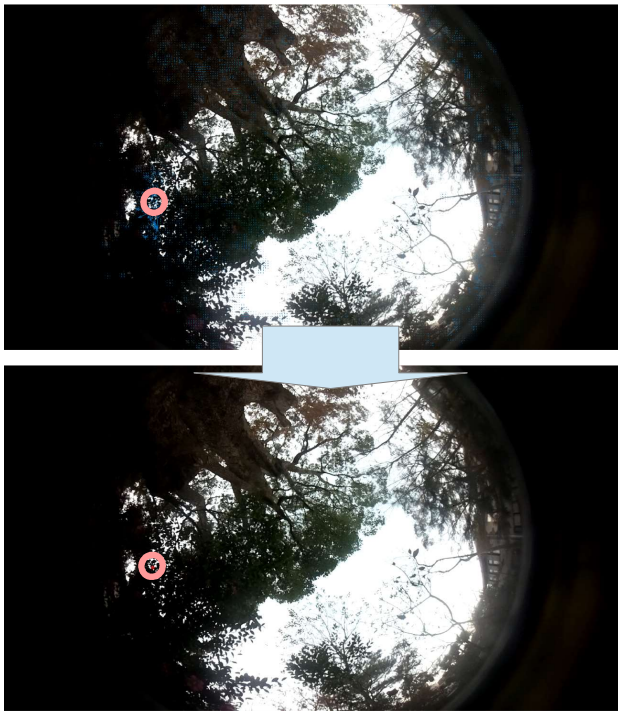


図 7: 鳥による枝の動きを検出

法を提案した．提案手法での課題は，背景となる木の風によるオプティカルフローと，鳥の行動によって生じたオプティカルフローを弁別する事であった．鳥の行動によって生じるオプティカルフローは，フローが生じる領域が鳥とともに移動するのに対し，風によるオプティカルフローは，ほぼ同じ位置で往復する．これを利用し，得られたオプティカルフロー画像に時空間的な平滑化をかけることで，風によるオプティカルフローを排除し，鳥の行動によるオプティカルフローを識別できるようになった．

現状，空間的平滑化のパラメータ σ_s，時間方向の平滑化パラメータ σ_t，二値化閾値 δ，領域重心の移動距離に対する閾値 θ_l，θ_h は，主に風による画像背景の動きに対して調整が必要である．今後の課題は，風速などの環境に適応したパラメータの自動調整，処理の効率化等が考えられる．

謝辞

本研究は JSPS 科研費 JP19KK0260 および JP20H00475 の助成を受けた．

参考文献

[1] K. P. Able and S. A. Gauthreaux. Quantification of nocturnal passerine migration with a portable ceilometer. *Condor*, 77(1):92–96, Jan. 1975

[2] I. Steiner, C. Bürgi, S. Werffeli, G. Dell'Omo, P. Valenti, G. Tröster, D. Wolfer, and H. P. lipp. A GPS logger and software for analysis of homing in pigeons and small mammals, *Physiol. Behav.*, 71(5):589–596, Dec. 2000.

[3] Z. Gayk and D. Mennill. Pinpointing the position of flyingnsongbirds with a wireless microphone array: Three dimensional triangulation of warblers on the wing. *Bioacoustics*, 29(4):1–12, May, 2019.

[4] R. Yoshihashi, R. Kawakami, M. Iida, and T. Naemura. Bird detection and species classification with time-lapse images around a wind farm: Dataset construction and evaluation. *Wind Anergy*, 20(12):1983–1995, Aug. 2017.

[5] A. Takeki, T. Trinh, R. Yoshihashi, R. Kawakami, M. Iida, andT. Naemura. Combining deep features for object detection at variousscales: Finding small birds inlandscape images, *IPSJ Trans.Comput. Vis. Appl.*, 8(5):1-7,Dec. 2016.

[6] E. Verreycken, R. Simon, B. Quirk-Royal, W. Daems, J. Barber, and J. Steckel. Bio-asoustic tracking and localization using heterogeneous scalable microphone arrays. *Common Biology*, 4(1275):1–11, NOv. 2021.

[7] S. Sumitani, R. Suzuki, T. arita, K. Nakadai, and H. Okuno. Non-invasive monitoring of the spatio temporal dynamics of vocalization among songbirds in a semi free flight environment using robot audition techniques. *Birds*, 2(2):158–172, April, 2021.

[8] Y. Kawanishi, I. Ide, B. Chu, C. Matsuhira, M. A. Kastner, T. Komamizu, and D. Deguchi, Detection of Birds in a 3D Environment Referrring to Audio Visual Information, *to appear in Proc. AVSS 2022*, Dec. 2022.

[9] Bruhn, Andrés, Joachim Weickert, and Christoph Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods.*International journal of computer vision*, 61.3 (2005): 211-231.

[10] Farneback, G. Two-Frame Motion Estimation Based on Polynomial Expansion. *In Proceedings of the 13th Scandinavian Conference on Image Analysis*, 363 - 370. Halmstad, Sweden: SCIA, 2003.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-04

PyHARK: HARKのオンライン・オフライン処理用Pythonパッケージ

PyHARK: HARK Python package supporting online and offline processing

中臺 一博[1*]　瀧ヶ平 将行[2]　糸山 克寿[1,2]
Kazuhiro NAKADAI[1]　Masayuki TAKIGAHIRA[1]　Katsutoshi ITOYAMA[2]

[1] 東京工業大学 工学院システム制御系
[1] Dept. Sys. & Contr. Eng., School of Engineering, Tokyo Institute of Technology
[2] （株）ホンダ・リサーチ・インスティチュート・ジャパン
[2] Honda Research Institute Japan Co., Ltd.

Abstract: 本稿では，ロボット聴覚オープンソースソフトウェア HARK 3.4 で新規に導入される PyHARK を HARK 講習会に先立ち紹介する。PyHARK は HARK の Python インタフェースを提供するパッケージであり，Python から HARK の機能のオンライン・オフライン呼び出しを可能にする実装である。そのアーキテクチャ，既存の HARK との違い，使い方を中心に解説する。

1　はじめに

筆者らは 2008 年のリリース以降，ロボット聴覚オープンソースソフトウェア HARK [1, 2, 3, 4, 5, 6, 7, 8, 9][1] の研究開発を続けている [10, 11, 12, 13, 14, 15]．本稿では， 2022 年 11 月 23 日にリリースを行う HARK の最新版 3.4 の新機能である PyHARK を紹介する．端的に言えば，PyHARK は，HARK の Python パッケージである．これまでにも HARK の開発環境上で Python プログラムとの連携を実現する HARK-Python が存在していたが，あくまでも HARK 上で Python プログラムを呼び出すための仕組みであった．また，HARK のプログラム開発では，HARK 独自のプログラミング環境 HARKDesigner を利用する必要があった．HARKDesigner は GUI プログラミング環境であるため，初心者には直感的でわかりやすいものの，Node.js[2] ベースであり，ブラウザを立ち上げて作業しなければならないため，プログラミングに慣れた人にとっては，必ずしも効率的な開発環境ではなかった．PyHARK では，HARK を Python プログラムから直接呼び出すことができるため，Python 開発によく利用される Jupiter Notebook[3] や Visual Studio Code[4] を使い，効率的なプログラミングが可能である．また，HARK の特長である逐次処理実行機能をそのまま継承しているので，Python で，センサやファイルからデータを逐次的に取得し処理を行うプログラムを作成することができる．ファイルをまとめて読み込んで処理をするオフライン・バッチ処理も実装可能である．

*東京工業大学 工学院システム制御系
〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: nakadai@ra.sc.e.titech.ac.jp

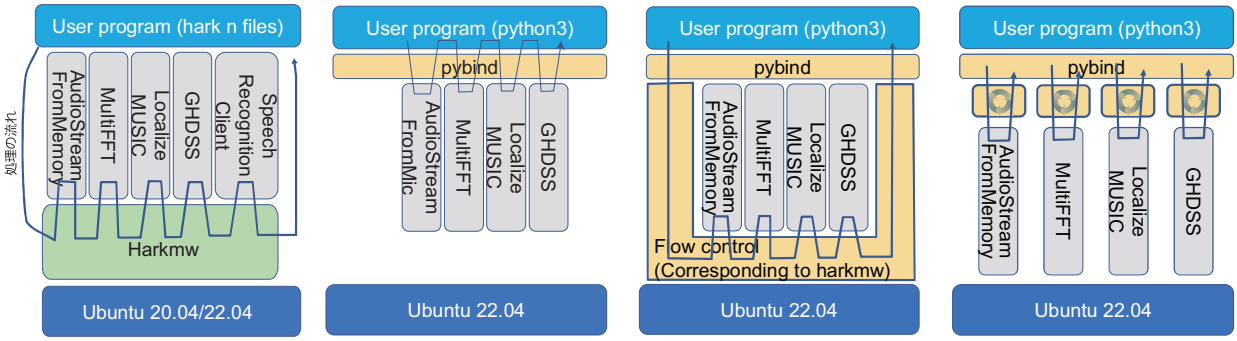
[1] https://hark.jp/
[2] https://nodejs.org/ja/
[3] https://jupyter.org/
[4] https://azure.microsoft.com/ja-jp/products/visual-studio-code/

2　HARK とその Python 化における課題

本節では，HARK から PyHARK に至るアーキテクチャを紹介し，PyHARK 化における課題，およびその解決法について議論する．

2.1　HARK のアーキテクチャ

図 1a) に従来版の HARK の代表として，HARK 3.4 のソフトウェアスタックを示す．従来版の HARK では，OS レイヤ (Ubuntu OS) の上に HARK の機能ノードの制御を司るミドルウェア harkmw [16]（緑色のボックス）が載っている．その上のレイヤは，harkmw を通じて動作する HARK の機能ノード（灰色のボックス群）で構成されている．最上位のレイヤはユーザプログラムレイヤ（水色のボックス）であり，このレイヤには，ユーザが，HARK の機能ノードを自由に配置し，それらの関係を記述することで作成した HARK のプログラムが置かれる．この HARK のプログラムは XML で記述されており，拡張子が “n” であるため n ファイルと呼ばれる．XML を直接手作業で記述することは現実的ではないため，HARKDesigner と呼ばれる独自の GUI プログラミング環境を用いて n ファイルを作成することで，プログラミングを行う．プログ



a) HARK 3.4　b) Python 化　c) PyHARK online　d) PyHARK offline

図 1: HARK のソフトウェアスタック

ラム実行時は n ファイルを harkmw に引数として与えて実行する．すると，harkmw は，n ファイルを読込み，XML の Parsing を行い，n ファイルの記述に沿って，フロー制御を行う．つまり，HARK の機能ノードは ユーザプログラムから直接呼び出されるのではなく，図 1 の矢印で示されているように harkmw から pybind11[5] を通じて，C/C++レベル の関数コールにより実行されることになる．このような仕組みを構築することにより，以下の 2 つの利点が得られる．

- ノード間統合のオーバーヘッドの低減：C/C++ レベル の関数コールで機能ノード間が接続されるため，統合のオーバヘッドが小さく，10 ms オーダの逐次処理が求められる音響信号処理でも十分な性能を発揮できる．
- 逐次処理記述の簡素化：ユーザが意識しなくても harkmw が裏で逐次処理を行うため，ユーザは逐次処理の 1 サイクル分の処理を記述するだけでよい[6]．この仕組みは，逐次処理が必要な時系列信号の処理全般に有効といえるので，機能ノードさえあれば，音響信号処理に限定せず利用することが可能である．

一方で，n ファイルを作成するためには独自の GUI 環境をわざわざ立ち上げてプログラミングしなければならないため，ある程度プログラミングに熟練したユーザには，プログラミングの作業効率が悪いという課題があった．

2.2 PyHARK に向けた課題

HARK の問題を解決するもっとも簡単な方法は，熟練プログラマでもストレスなく利用できるプログラミング言語・環境で HARK を利用できるようにすることである．近年は，信号処理や機械学習では，Python が一般的に利用されるようになってきている．また，Jupiter notebook や Visual Studio Code など Python をサポートし，かつインタラクティブにプログラミングやデバッグができる高機能なプログラミング環境が登場している．そこで，HARK を Python から利用できるよう HARK の Python パッケージ化を図ったのが PyHARK である．PyHARK をもっとも簡単に実現するには，図 1b) に示すように，HARK の機能ノードを，pybind11 でラップし，ユーザには，Python の関数として見えるようにすることである．このようにすれば．Python プログラム内で，”import hark” を記述することで，Python 内で HARK の機能が利用できるようになる．これだけ考えると，もはや，harkmw のようなミドルウェアは不要に見えるが，実際には，このままでは，以下の 2 つの問題が発生してしまう．

- ノード間統合のオーバーヘッドの増大
- オフライン・バッチ的な使い方は実装が大変

一つ目の問題は，harkmw がない場合，機能ノード間に直接のつながりはないため，機能ノード間のデータの受け渡しをユーザプログラム（ Python）レイヤで行わざるを得ないことに起因している．これは，音響信号のような時系列信号を扱う場合は特に大きな問題となる．音響信号処理では，一般的に，一連のデータをある時間単位で分割（この単位を一般にフレームという，音響信号の場合は 10〜50 ms とすることが多い）し，フレーム単位で逐次的に処理を行う．例えば，図 1b) のように，AudioStreamFromMic, MultiFFT, LocalizeMUSIC, GHDSS という機能ノードを順番に実行する処理を考える．あるフレームのデータをある機能ノードから次の機能ノードに渡す際（例えば，MultiFFT から LocalizeMUSIC に渡す場合）に，そのデータをいちいちユーザプログラムレイヤまで引き上げてから機能ノードレイヤに戻すというレイヤをまたいだ通信が必要になる．このレイヤをまたぐ通信が一度行われるたびに，データのシリアライズ・デシリアライズが発生する．図 1b) の矢印に示すように，このレイヤをまたぎの通信は AudioStreamFromMic から GHDSS まで

[5] https://github.com/pybind/pybind11
[6] Perl で暗黙のループを実現する ”-n” スイッチと似たイメージ

の一連の処理で機能ノードにデータが渡されるたびに実行されることになる．さらに，この一連の処理がすべてのフレームについて行われるため，必然的にオーバヘッドが大きくなる．

二つ目の問題は，機能ノードが，もともとフレーム単位で処理を行うように作られていることに由来する．従来の HARK では，これはフレーム単位の逐次処理を可能とする利点であったが，図 1b) に示すような構成をとった場合，従来の HARK では，harkmw が隠ぺいしてくれていた逐次処理に相当する，入力信号をフレーム単位に分割し，１フレームごとに機能ブロックを呼び出すコードをユーザ側で用意しなければならなくなってしまう．このため，従来の HARK に比べて使い勝手が悪くなってしまう．

3　PyHARK アーキテクチャ

PyHARK では, 逐次処理, オフライン・バッチ処理の二種類に処理を分けて考えることで HARK の Python 化における２つの課題の解決を図っている．また，詳細は割愛するが，HARK の C++ 部分の実装を見直し，マルチスレッド化，高速な行列演算を提供するライブラリである Eigen を導入によることによって，オリジナルの OSS 版と比較して，高速化を行っている．

3.1　逐次版 PyHARK

図 1c) に逐次版の PyHARK の構成図を示す．逐次版では，従来の HARK の利点である逐次処理を継承できるよう，機能ノードのフロー制御を導入し，ユーザの Python プログラムから呼び出しができるようになっている．フロー制御部分は，基本的には，harkmw から n ファイルの読込み，Parsing 機能を取り除いたものに相当している．また，従来版では，harkmw は実行ファイルとなっていたが，PyHARK ではユーザのプログラム上で，Publisher, Subscriber を呼び出すことで実現している．このような違いはあるものの，端的に言えば，これまでの n ファイルに相当する部分を python プログラムとして記述できるようにしたパッケージと捉えることができる．図 1c) の矢印で示すように実行時のフロー制御は，フロー制御部にまかせて，内部的には従来版 HARK と同様，C/C++レベルの関数コールでの機能ノード接続となっているため，オーバヘッドは従来版と同様に低く抑えることができる．もちろん，Python でコーディングした自前の機能ノードを用いる場合や機能ノードの途中経過を probe したい時などは pybind11 を経由した機能ノードへのアクセスも可能であるが，この場合は，シリアライズ・デシリアライズに伴う通信のオーバヘッドが発生する．

3.2　オフライン・バッチ処理版 PyHARK

図 1d) にオフライン・バッチ処理版の PyHARK の構成図を示す．逐次版では，逐次処理が可能な代わりに, n ファイルに相当する機能ノードや機能ノード間の接続ネットワークの定義を Python プログラムとして記載する必要があった．HARKDesigner を使わずに記述できる反面，これまで XML として記述していたものと同様の内容をプログラム化しなければならないので煩雑である．また，実際に Python でプログラミングする場合, 大抵はファイルもしくはファイルセット単位で順番に処理を行うオフライン・バッチ処理として記述することが多い．オフライン・バッチ処理を念頭に置いている場合にも逐次処理 のプログラムを書かなければならいのでは使い勝手が悪い．深層学習のパッケージとしてよく用いられる Keras では，すべてのデータを一度に読込んで学習できる場合は，fit() を使って簡単に学習コードを記述できる．しかし，データ量が多くなり，メモリ上に一度に読み込めない場合は，逐次的にデータを読みながら学習を実行する fit_generator() を別途実装する必要があり，実装が煩雑になる[7]．誤解を招くことを恐れずに書けば，PyHark の逐次版は，常に fit_generator() に相当する実装を強いられるのと近いイメージといえる．そこで，PyHARK では，逐次処理を一切意識せず，オフライン・バッチ処理的に Python のプログラミング作成することも可能である．これを実現するため，HARK の Python 化における課題の 2 つ目として挙げている機能ノードが，もともとフレーム単位で処理を行うように作られているということをユーザが意識せずに済むようにする作りになっている．具体的には，pybind11 経由で機能ノードを呼び出す際に，フレームごとに回して処理をする部分を自動的に行うような実装を行っている．当然，シリアライズ・デシリアライズは発生するものの，そもそもオフライン・バッチ処理用の仕組みであるため，通信のオーバヘッドが問題にはならない場合の構成である（問題になる場合は逐次版を使えばよい）．

4　PyHARK を用いた実装例

PyHARK を用いて音源定位を行うプログラムの例を Listing 1（オンライン版）と 2（オフライン版）に示す．これらは，短時間フーリエ変換，MUSIC 法による音源定位，音源追跡処理を行うプログラムで，実装的には，主に，HARK の MultiFFT, LocalizeMUSIC, SourceTracker ノードを用いている．

Listing 1: pyhark-online-sample.py

```
1 #! /usr/bin/env python
2 # -*- coding: utf-8 -*-
```

[7]現在は fit_generator() は fit() に統合されている

```
3
4  import sys
5  import threading
6  import time
7  import numpy
8  import soundfile
9  import hark
10 import plotQuickSourceKivy
11
12 # マイク数等設定
13 nch    = 8
14 winlen = 512
15 advance = 160
16
17 # ネットワークの定義HARK(Localize)
18 class HARK_Localize(hark.NetworkDef):
19     def build(self,
20               network: hark.Network,
21               input:   hark.DataSourceMap,
22               output:  hark.DataSinkMap):
23
24         # 機能ノードの生成
25         node_audio_stream_from_memory = network.
             create(hark.node.AudioStreamFromMemory)
26         node_multi_fft = network.create(hark.node.
             MultiFFT)
27         node_localize_music = network.create(hark.
             node.LocalizeMUSIC)
28         node_cm_identity_matrix = network.create(
             hark.node.CMIdentityMatrix, dispatch=hark
             .RepeatDispatcher)
29         node_constant = network.create(hark.node.
             Constant, dispatch=hark.RepeatDispatcher)
30         node_source_tracker = network.create(hark.
             node.SourceTracker)
31         node_plotsource_kivy = network.create(
             plotQuickSourceKivy.plotQuickSourceKivy)
32
33         try:
34             # 機能ノードのプロパティ設定とノード間接続
35             r = [
36                 node_audio_stream_from_memory
37                     .add_input("INPUT", input["
                       INPUT"])
38                     .add_input("CHANNEL_COUNT", nch
                       )
39                 ,
40                 node_multi_fft
41                     .add_input("INPUT",
                       node_audio_stream_from_memory
                       ["AUDIO"])
42                 ,
43                 node_cm_identity_matrix
44                     .add_input("NB_CHANNELS", nch)
45                     .add_input("LENGTH", winlen)
46                 ,
47                 node_constant
48                     .add_input("VALUE", True)
49                 ,
50                 node_localize_music
51                     .add_input("INPUT",
                       node_multi_fft["OUTPUT"])
52                     .add_input("NOISECM",
                       node_cm_identity_matrix["
                       OUTPUT"])
53                     .add_input("OPERATION_FLAG",
                       node_constant["OUTPUT"])
54                     .add_input("A_MATRIX", "tf.zip"
                       )
55                     .add_input("MUSIC_ALGORITHM", "
                       SEVD")
56                     .add_input("WINDOW_TYPE", "PAST
                       ")
57                     .add_input("
                       LOWER_BOUND_FREQUENCY", 500)
58                     .add_input("
                       UPPER_BOUND_FREQUENCY", 2800)
59                     .add_input("WINDOW", 50)
60                     .add_input("PERIOD", 1)
61                     .add_input("NUM_SOURCE", 2)
62                 ,
63                 node_source_tracker
64                     .add_input("INPUT",
                       node_localize_music["OUTPUT"
                       ])
65                     .add_input("THRESH", 25)
66                     .add_input("PAUSE_LENGTH",
                       1200.0)
67                     .add_input("MIN_SRC_INTERVAL",
                       20.0)
68                     .add_input("MIN_ID", 0)
69                 ,
70                 node_plotsource_kivy
71                     .add_input("SOURCES",
                       node_source_tracker["OUTPUT"
                       ])
72                 ,
73             ]
74
75             # 出力設定
76             output.add_input("OUTPUT",
                 node_source_tracker["OUTPUT"])
77
78         except BaseException as ex:
79             print('error: {}'.format(ex))
80
81         return r
82
83 # ネットワークの定義HARK(ループMAIN)
84 class HARK_Main(hark.NetworkDef):
85     def __init__(self):
86         hark.NetworkDef.__init__(self)
87
88     def build(self,
89               network: hark.Network,
90               input:   hark.DataSourceMap,
91               output:  hark.DataSinkMap):
92
93         try:
94             # フロー制御用
95             node_publisher = network.create(hark.
               node.PublishData, dispatch=hark.
               RepeatDispatcher, name="Publisher")
96             node_subscriber = network.create(hark.
               node.SubscribeData, name="Subscriber"
               )
97             # フレーム毎音源定位処理
98             loop = network.create(HARK_Localize,
               name="HARK_Localize")
99         except BaseException as ex:
100            print(ex)
101
102        # フロー制御との接続
103        r = [
104            loop.add_input("INPUT", node_publisher[
               "OUTPUT"]),
105            node_subscriber.add_input("INPUT", loop
               ["OUTPUT"])
106        ]
107
108        return r
109
110 # 結果取得用
111 def received(data):
112     print('>>>> received: {}'.format(data))
113
114 def main(args=sys.argv[1:]):
115
116     # ネットワーク読込みHARK
117     network = hark.Network.from_networkdef(
               HARK_Main, name="HARK_Main1")
118
119     # フロー制御用
120     publisher = network.query_nodedef("Publisher")
121     subscriber = network.query_nodedef("Subscriber"
               )
122
123     # 結果取得用
124     subscriber.receive = received
125
126     # 読込んだネットワークの実行
127     try:
128         def target():
129             network.execute()
130
131         th = threading.Thread(target=network.
               execute)
132         th.start()
133     except BaseException as ex:
134         print(ex)
135
136     # 音響信号読込み (8 ch, 16bit integer)
137     audio, rate = soundfile.read('input.wav', dtype
               =numpy.int16)
138
139     # シフト長 advance(160) でフレーム化
140     frames = numpy.lib.stride_tricks.
               sliding_window_view(audio.T, (nch, advance))
               [0, ::advance]
141
142     # フレーム毎実行
143     try:
144         for t in range(frames.shape[0]):
```

```
145             if not th.is_alive():
146                 break
147             print('<<<<␣send:␣count={}'.format(t))
148             publisher.push(frames[t,:,:])
149             time.sleep(0.01)
150     finally:
151         publisher.close()
152         network.stop()
153         th.join()
154
155 if __name__ == '__main__':
156     main(sys.argv[1:])
```

Listing 2: pyhark-offline-sample.py

```
1  #! /usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4  import hark
5  import numpy
6
7  # マイク数等設定
8  nch    = 8
9  winlen = 512
10 advance = 160
11
12 # 音響信号読込み (8 ch, 32 bit float)
13 audio, rate = soundfile.read('input.wav', dtype=
   numpy.float32)
14
15 # フレーム化フレーム長( 512 シフト長 160)
16 frames = numpy.lib.stride_tricks.
   sliding_window_view(audio, winlen, axis=0)[::
   advance, :, :]
17
18 multi_fft = hark.node.MultiFFT()
19 multifft_out = multi_fft(INPUT=frames)
20
21 noisecm = numpy.eye(nch, dtype=numpy.complex64).
   flatten()
22 noisecm_bins = numpy.broadcast_to(noisecm, (
   multifft_out.OUTPUT.shape[0], multifft_out.OUTPUT
   .shape[1], nch*nch))
23 localizemusic_node = hark.node.LocalizeMUSIC()
24 localizemusic_out = localizemusic_node(INPUT=
   multifft_out.OUTPUT, NOISECM=noisecm_bins,
   A_MATRIX='tf.zip', MUSIC_ALGORITHM='SEVD', PERIOD
   =1, WINDOW_TYPE="PAST", WINDOW=50, NUM_SOURCE=2,
   LOWER_BOUND_FREQUENCY=500, UPPER_BOUND_FREQUENCY
   =2800,ENABLE_OUTPUT_RXXN=True)
25
26 sourcetracker_node = hark.node.SourceTracker()
27 sourcetracker_out = sourcetracker_node(INPUT=
   localizemusic_out.OUTPUT, THRESH=25.0,
   PAUSE_LENGTH=1200.0,MIN_SRC_INTERVAL=20.0,MIN_ID
   =0)
28
29 print(sourcetracker_out.OUTPUT)
```

Listing 1 では，n ファイルに相当するネットワークの定義を行っているのが，HARK_Main クラス (83–108 行目)，HARK_Localize クラス (17–81 行目) である．HARK_Main クラスでフロー制御を含めたネットワークの枠組みを用意し，1 フレーム分の音源定位処理を記述した HARK_Localize クラスがその中で展開される形になっている．これらのクラスの中では，用いる機能ノードの宣言，各機能ノードのプロパティ設定，機能ノード間の接続設定が記述されている．HARK_Main では，これに加え，フロー制御とのインタフェース用に，Publisher, Subscriber に関する記述を行っている．114–153 行目の main 関数では，上記のクラスとして定義されているネットワーク定義を読込み，スレッドとしてこれを実行する．その後，フレームごとのデータを publisher を通じて push している (148 行目)．PyHARK 内部では，決められたフレーム長を単位として，逐次処理が実行される（何も指定しなければデフォルト値として 512 サンプルがフレーム長として用いられる）．このプログラムでは，137 行目で読み込んだ音響データを 140 行目でフレームシフト長である 160 サンプルごとにフレーム化を行っているが，実センサでは必ずしもフレームシフト長分のデータが毎回きっちり得られるわけではない．実際には，HARK ネットワーク内の AudioStreamFromMemory (25 行目) がバッファ処理を行い，取得データ量の揺れを吸収する仕組みになっているので，取得した分だけ push すればよい仕組みになっている．

Listing 2 は，逐次処理を意識する必要がないので，事前にネットワーク定義を行う必要もなく，逐次版に比べシンプルに記述することができる．13 行目は，入力信号を読み込む部分であり，データ型を float32 として読み込んでいることを除けば，逐次版と同等のコードになっている[8]．逐次版では，フレームシフト長分 (センサの場合は，センサからその時までに得られたデータのみ）を publisher に push しており，512 サンプルのフレームを構成する処理は，AudioStreamFromMemory が行っていた．オフライン・バッチ版では，最初からすべてのデータが利用可能であることが前提であるので，わざわざこのような処理をするまでもなく，16 行目で直接，全データに対して，フレーム長 512 サンプル，シフト長 160 サンプルでフレーム化処理を行っている．フレーム化したデータをまとめて multi_fft に入力すれば，STFT の結果がまとめて得られ，その結果を localizemusic_node に入力すれば，定位結果を，さらにその結果を sourcetracker_node に入力すれば音源追跡結果を得ることができる．オフライン処理で記述すれば，このように，直感的に記述することが可能である．

PyHARK を用いれば，プロトタイピングの際は，オフライン・バッチ版を用いて記述し，その後逐次版に移行することで，同じ Python 上で実センサを用いて逐次処理版のプログラムを比較的容易に構築することができる．また，現在計画を進めている組込版は逐次処理版と親和性が高い設計になっているので，IoT などの実開発への移行コストを低減することが可能と考えられる．

5 評価実験

Listing 1，Listing 2，および従来の HARK の処理速度の比較を行った．

input.wav として，8 チャンネルの音響信号 20 秒分を用いた．この信号には，マイクロホンアレイからみて，0 度と 180 度の方向に，それぞれ白色雑音が音源として含まれている．実験は，VMWare Player 16 上の Ubuntu 22.04 OS で行った．VM には Intel i7-12700K

[8] 逐次版とのデータ型の不整合については今後改善する予定

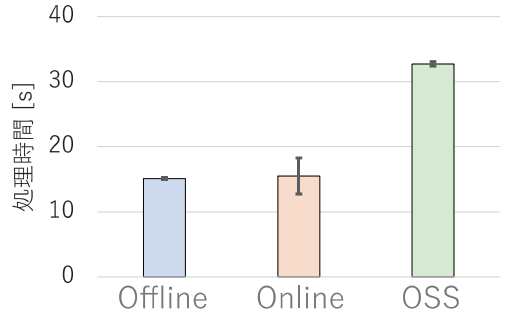


図 2: 処理時間計測結果

を 4 つ，またメモリを 32GB 割り振った．また，実験の際は，表示処理などの I/O の影響をなるべく低減するために，kivy などの描画や標準出力に書き込む処理を OFF にして実験を行った．また，従来は，実時間処理を担保するために，LocalizeMUSIC の固有値展開実行の頻度を通常は 50 フレームに一回と間引いて実行 (PERIOD=50) していたが，今回は PERIOD の値を 1 に設定することで毎フレーム固有値展開を実行する設定とした．実験は，各条件ごとに 10 回ずつ行い，平均と標準偏差をプロットした．

結果を図 2 に示す．まず，オフライン処理，オンライン処理共に，信号長である 20 秒以下で処理が終わっており，毎フレーム固有値展開を行っても，実時間処理性が保たれていることがわかる．また，オンライン処理には，実行時間にばらつきがあるものの，両者とも，平均実行時間は同程度となっており，オーバヘッドが同等であることがわかる．一方で，従来の HARK はリアルタイムファクターで 1.5 以上となっている．実際には，従来の HARK と比較すれば，PyHARK の実装のオーバヘッドは若干ではあるが，大きくなっているはずである．この結果は，PyHARK の実装の際に，同時に行った C++ 部分のマルチスレッド化，および Eigen の導入が，増加したオーバヘッド以上に効いており，高速化が実現できたと考える．

6 おわりに

本稿では，ロボット聴覚オープンソースソフトウェア HARK 3.4 について新たに加わった新機能である PyHARK に焦点を絞って紹介した．PyHARK は HARK の機能を Python から利用できるようにすると同時に，従来の HARK の利点である逐次処理も記述可能なパッケージである．設計的には，RaspberryPi などの ARM ベースの組込みシステム，FPGA，GPU への移植も考慮に入れた設計となっているため，将来的には，プロトタイプから実開発までシームレスに移行することが可能である．今後は，PyHARK の安定化，従来版 HARK とのソースの統合，ARM, FPGA, GPU サポートの検討を進めていきたい．

謝辞

本稿にかかる研究の一部は，JSPS 科研費 JP19KK0260 および JP20H00475 の助成を受けた．

参考文献

[1] K. Nakadai, T. Takahashi, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino. Design and implementation of robot audition system ”HARK”. *Advanced Robotics*, Vol. 24, pp. 739–761, 2010.

[2] K. Nakadai, H. G. Okuno, and T. Mizumoto. Development, deployment and applications of robot audition open source software HARK. *Journal of Robotics and Mechatronics*, Vol. 29, No. 1, pp. 16–25, 2017.

[3] 中臺一博. オープンソースコミュニティーに貢献するということ. 映像情報メディア学会誌, Vol. 71, No. 5, pp. 647–653, 2017.

[4] 中臺一博, 奥乃博. ロボット聴覚用オープンソースソフトウェア HARK の展開. デジタルプラクティス, Vol. 2, No. 2, pp. 133–140, 2011.

[5] K. Nakadai, H. G. Okuno, T. Takahashi, K. Nakamura, T. Mizumoto, T. Yoshida, T. Otsuka, and G. Ince. Introduction to open source robot audition software HARK. In *The 29th Annual Conference of the Robotics Society of Japan (RSJ2011)*, 2011.

[6] 奥乃博, 中臺一博. ロボット聴覚オープンソフトウエア HARK. 日本ロボット学会誌 特集「ロボット聴覚」, Vol. 28, No. 1, pp. 6–9, 2010.

[7] K. Nakadai, H. G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino. An open source software system for robot audition hark and its evaluation. In *2008 IEEE RAS International Conference on Humanoid Robots (Humanoids 2008)*, pp. 561–566, 2008.

[8] 中臺一博, 奥乃博, 中島弘史, 長谷川雄二, 辻野広司. ロボット聴覚オープンソースソフトウェア HARK の概要と評価. 第 26 回日本ロボット学会学術講演会予稿集 (RSJ 2008), 2008.

[9] 中臺一博, 山本俊一, 奥乃博, 中島弘史, 長谷川雄二, 辻野広司. ロボット聴覚用オープンソースソフトウェア HARK の概要. ロボティクス・メカトロニクス 講演会 2008 講演論文集, 2008.

[10] 公文誠, 若林瑞穂, 干場功太郎, 中臺一博, 奥乃博. ドローンによる地上音源の位置推定 - HARK を用いたドローン聴覚の取り組み. 第 19 回計測自動制御学会システムインテグレーション部門講演会 (SI2018) 講演論文集, 2018.

[11] 鈴木麗璽, 炭谷晋司, 中臺一博, 奥乃博. ロボット聴覚技術を用いた鳥類の歌行動分析の試み - 複数のマイクロホンアレイを用いた二次元リアルタイム歌定位 -. 第 18 回計測自動制御学会システムインテグレーション部門講演会 (SI2017) 講演論文集, pp. 1124–1126, 2017.

[12] 中臺一博, 坂東宜昭, 水本武志, 干場功太郎, 小島諒介, 糸山克寿, 杉山治, 公文誠, 奥乃博. HARK 2.3 の紹介とタフロボティクスチャレンジへの展開. 第 17 回計測自動制御学会システムインテグレーション部門講演会 (SI2016) 講演論文集, pp. 2175–2178, 2016.

[13] 中臺一博, 水本武志, 中村圭佑, 奥乃博. HARK 2.2 の新機能とその組込み，saas への展開. 第 16 回計測自動制御学会システムインテグレーション部門講演会 (SI2015) 講演論文集, pp. 1835–1838, 2015.

[14] 中臺一博, 奥乃博. ロボット聴覚オープンソースソフトウェア HARK の紹介. 第 15 回計測自動制御学会システムインテグレーション部門講演会 (SI2014) 講演論文集, pp. 1712–1716, 2014.

[15] 中臺一博, 奥乃博. ロボット聴覚用オープンソースソフトウェア HARK 1.0.0 の概要. 第 11 回計測自動制御学会システムインテグレーション部門講演会 (SI2010) 講演論文集, pp. 1771–1774, 2010.

[16] 木下智義, 中臺一博. ロボット聴覚オープンソースソフトウェア hark 用ミドルウェア hark middleware の紹介. 人工知能学会研究会資料 SIG-Challenge-057-012, pp. 73–78, 2020.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-05

スマートグラスを用いた音環境理解支援

Assistance of Sound Scene Understanding with Smart Glasses

吉井 和佳 [1,2*]
Kazuyoshi Yoshii[1,2]

[1] 京都大学 大学院情報学研究科　[2] 理化学研究所 革新知能統合研究センター (AIP)
[1]Graduate School of Informatics, Kyoto University　[2]AIP, RIKEN

Abstract: 本稿では，実環境下での音声コミュニケーションを支援するため，特定の話者の音声をリアルタイムで強調・認識・翻訳・拡張現実 (AR) 表示を行うスマートグラスの開発について紹介する．スマートグラスに搭載されたマイクアレイから得られる音響信号は，未知の環境における多数の音声・雑音・残響が重畳したものであり，音源の移動に加えて，ユーザ自身の移動や頭部の動きに影響を受けるため，研究室環境のベンチマークとは本質的に難しさが異なる．そのうえで，頑健性と即応性を備えた実用システムを開発する技術的なチャレンジについて解説する．

1　はじめに

近年，スマートフォンやスマートスピーカーへの指示，自動字幕・議事録作成など，日常生活中で音声認識が活用される場面が増えつつある．最新の深層学習技術に基づく音声認識手法では，接話型マイクで集音されたクリーンな発話に対する認識精度は 90%を超えており，実用化の急速な進展を支えている．一方，雑音・残響のある実環境における遠隔発話認識となると，認識精度が 30%以下に低下することも珍しくない．そのため，遠隔音声認識のフロントエンドとして，マイクアレイで集音された多チャネル音響信号に対する音声強調や残響除去の研究が活発である．

我々の研究グループでは，汎用的な基盤技術として，多チャネル音源分離・音声強調技術の改善に継続的に取り組んできた（第 2 章）．具体的には，高速多チャネル非負値行列因子分解 (FastMNMF) [1,2] をコア技術として，深層学習の導入 [3–6]，残響除去の統合 [7,8]，音源数の自動推定 [9]，優ガウス性音源への対応 [10] を考案した．一連の技術は，遠隔音声認識のフロントエンドとして有用であり，アンドロイドロボットによる音声対話への応用を想定している．

将来有望な別の応用として，健常者・聴覚障碍者双方に有用な，スマートグラスを用いた音声コミュニケーション支援に取り組んでいる（第 3 章）．具体的には，マイクアレイ・カメラを用いて視聴覚情報を取得し，特定話者の音声を強調したうえで，リアルタイムで音声認識・翻訳結果を拡張現実 (AR) 表示する [11–13]．しかし，センサー類はウェアラブルデバイスに搭載されているため，ユーザの移動や頭部の動きに対する頑健性に加えて，リアルタイム性（低遅延）も求められ，研究室環境における伝統的なオフライン型のベンチマークとは本質的に難しさが異なる．

本稿では，多チャネル音源分離における我々の貢献を体系的に解説し，スマートグラスへの応用に関する先駆的取り組みを通じて今後を展望する（第 4 章）．

*連絡先：京都大学 大学院情報学研究科 知能情報学専攻
〒 606-8501 京都府京都市左京区吉田本町
E-mail: yoshii@i.kyoto-u.ac.jp

2　多チャネル音源分離・残響除去

我々の研究グループは，汎用ブラインド音源分離 (BSS) の洗練化に継続的に取り組み，理論面で一定の完成をみたと言ってよい．これまで，BSS 問題においては，観測音の生成モデルに基づく統計的アプローチが広く用いられてきた．中でも，多チャネル非負値行列因子分解 (MNMF) [14] は，フルランク空間共分散行列に基づく空間モデルと，非負値行列因子分解 (NMF) に基づく音源モデルを統合した汎用 BSS 手法として注目されている．しかし，計算負荷が過大であり，局所解に陥りやすい欠点があった．独立低ランク行列分析 (ILRMA) [15] では，空間共分散行列をランク 1 に制限することで，計算量を削減し，初期値依存性を軽減しているが，残響に対する頑健性が低下する代償があった．

最近我々が提案した，高速かつ高精度な BSS 手法である高速多チャネル非負値行列因子分解 (FastMNMF) [1,2] では，空間共分散行列のフルランク性を保ちつつも，同時対角化制約を導入することで，計算量の削減と分離精度の改善を両立することに成功した．さらに，FastMNMF 各部の本質的な拡張を行い，確率的な枠組みへの深層学習の導入を実現した（図 1）．

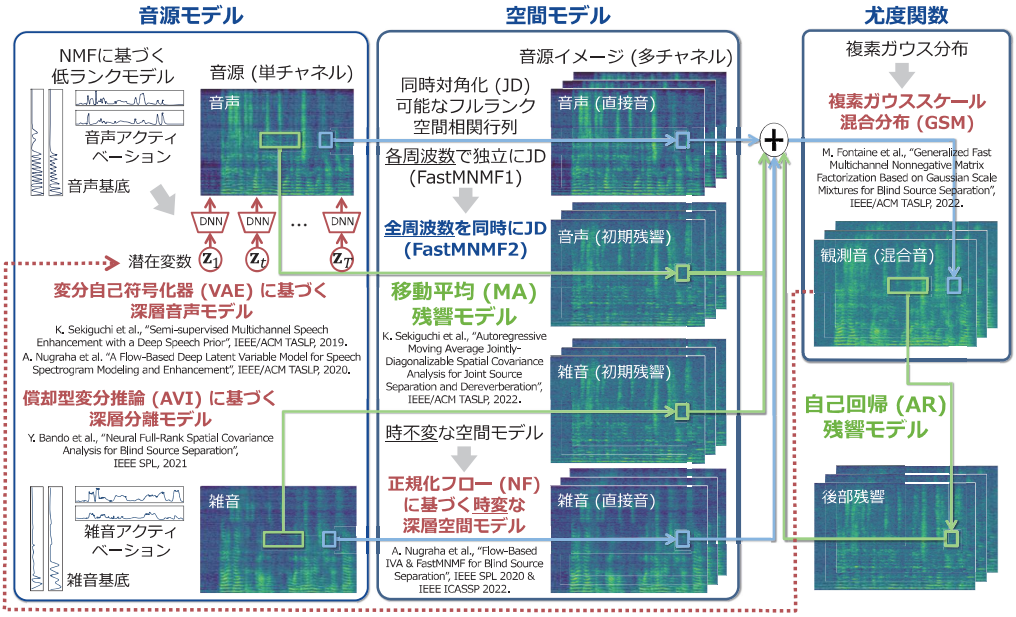


図 1: FastMNMF を中心とした多チャネル音源分離技術体系

2.1 フルランク空間共分散分析の系譜

まず，フルランク空間共分散行列に基づく BSS 手法の歴史を整理しておく．一連の研究の発端となったのは，Duong ら [16] が考案した，音源の時間周波数平面上のパワースペクトル密度を表現する音源モデルと，マイク間の相関（位相差）を表現する空間モデルからなるフルランク空間共分散分析 (FCA) である．FCA は統一的な確率モデルの最尤推定に対応しており，数学的に見通しが良く，汎用性が高い一方で，音源・空間モデルの自由度が高すぎるゆえに，物理的に妥当なパラメータを推定することは容易ではなかった．

澤田ら [14] は，低ランク音源モデルに基づく多チャネル非負値行列因子分解 (MNMF) を考案した．MNMF では，音源のパワースペクトル密度は非負値行列因子分解 (NMF) で表現されており，パラメータの最尤推定に収束保証付きの Majorization-Minimization (MM) アルゴリズムを用いた点で画期的である．しかし，コスト関数（負の対数尤度）の上限関数を導出する過程に誤りがあり，フルランク空間共分散行列の更新則はリッカチ方程式の解として与えられていた．我々はその後，NMF の本質的なテンソル拡張である相関テンソル分解 (CTF) [17] およびその高速版 (FastCTF) [18] の研究において，上限関数の導出を正しく行ったうえで，上記リッカチ方程式の解が，その係数である二つの正定値行列の幾何平均で与えられることを報告した．この閉経式での更新則は，後に続く研究で標準的に用いられるようになっている．MNMF は優れた分離性能を達成しうるが，空間モデルの自由度は依然として高く，局所解へ陥りやすい問題が残されていた．

近年，奇しくも同時期に，我々の研究グループ [1,2] と伊藤らの研究グループ [19,20] から独立に，音源の空間共分散行列を同時対角化可能なものに制限した高速多チャネル非負値行列因子分解 (FastMNMF) が提案されている（名称・確率モデルともに同一）．FastMNMF は最先端の BSS 手法であり，空間モデルの自由度を制限しつつもフルランク性を維持することにより，独立低ランク行列分析 (ILRMA) [15] に匹敵する計算効率と，MNMF より優れた分離性能を兼ね備えている．伊藤らは当時すでに，同時対角化制約を導入した FCA である FastFCA を考案していた [21,22]．同時期に，我々は，CTF の計算量を削減するための同時対角化制約の導入を考案していた [17,18]．そのため，同時対角化制約の MNMF への導入は自然な流れでもあった．

パラメータの最尤推定には共通して MM アルゴリズムが利用されているが，その実装は異なっていた．空間共分散行列を対角化する行列を更新するため，我々は，独立ベクトル分析 (IVA) [23] を始め，ILRMA [15] や FastCTF [18] と同様に，反復射影法 (IP) が適用できることを見出したのに対して [1]，伊藤らは不動点反復法を利用していた [19]．収束性が保証され，収束も速いことから，現在では IP の利用が標準的である．

我々は最終的に，FastMNMF1 [2,19,20] より性能に優れた FastMNMF2 [2] を考案した．FastMNMF1 では，周波数ごとに空間共分散行列を同時対角化可能なものに制限するのに対して，FastMNMF2 では，すべての周波数・音源に関する空間共分散行列を一挙に制限する点で異なる．これにより，各音源の方向情報がすべての周波数間で共有され，周波数間のパーミュテーション問題の解決に寄与することが期待できる．

2.2 深層音源モデルの導入 [3]

FastMNMF では，各音源スペクトログラムのパワースペクトル密度が低ランク構造を持つと仮定しているため，ある種の音源（例 : 音声スペクトログラム）に対してこの仮定は成り立たなかった．そこで，雑音環境下での音声強調を目的として，音声に対しては変分自己符号化器 (VAE) に基づく深層生成モデル [24] を用い，雑音に対しては NMF に基づく低ランクモデルを用いた音声強調法 FastMNMF-DP を提案した．VAE はあらかじめ事前学習しておくことが想定されるが，理論上は教師なし学習も可能な枠組みとなっている．

2.3 深層空間モデルの導入 [4, 6]

独立成分分析 (ICA) [25] や独立ベクトル分析 (IVA) [26] などの線形時不変型決定系 BSS 手法は，マイク数と音源数が等しい決定条件のもとで，周波数ごとに，混合音スペクトルを統計的に独立な要素からなる音源スペクトルに変換するための時不変な分離行列を推定する．この種の BSS では，分離行列と混合行列は逆行列の関係にあり，混合音スペクトルと音源スペクトルは可逆な確率変数である．このような確率変数の可逆変換は，正規化フロー (NF) の一種であるとみなせる [4]．そこで，FastMNMF の空間モデルに含まれる対角行列（空間変換行列）を NF で表現することで，音環境の変化（音源移動）にも対応可能な線形時変非決定系 BSS である NF-FastMNMF の導出に成功した．

2.4 残響モデルの導入 [7, 8]

実環境下において，音声強調と残響除去はいずれも重要であり，相互依存関係のある両タスクを一挙に実行することが望ましい．そこで，自己回帰移動平均 (ARMA) 過程に基づく残響モデルを FastMNMF に統合することで，音源分離と残響除去を高速かつ一挙に行う ARMA-FastMNMF を提案した．具体的には，NMF に基づく音源モデルに従う各音源信号に対し，移動平均 (MA) 過程に従って初期残響（音源に依存）が付加され，それらが重畳した混合信号に対し，AR 過程に従って後部残響（音源に非依存）が付加される生成モデルを考案し，混合音からの教師なし学習を実現した．

2.5 裾の重い尤度関数の導入 [10]

通常，混合音スペクトルに対する尤度関数として，時変な空間共分散行列をもつ複素ガウス分布を用いるのが一般的であり，突発性音源に対しては性能が劣化しやすい問題があった．そこで，裾の重い分布であるガウススケール混合分布 (GSM) を用いた GSM-FastMNMF を考案し，EM アルゴリズムに基づく統一的なパラメータ推定法を導出した．これまで独立に提案されてきた様々な裾の重い分布（t 分布 [27]，裾が重い一般化ガウス分布 (GG) [28]，α-対称安定分布 [29] など）に基づく FastMNMF を統一的に説明することに成功し，一般化双曲型分布 (GH) やその特殊系である正規逆ガウス分布 (NIG) が高い性能を示すことを発見した．

2.6 深層分離モデルの導入 [5]

通常，深層学習を用いた音源分離を行うには，混合音と音源信号とのペアデータから分離モデルを教師あり学習する必要があったが，実環境を網羅的にカバーする学習データを集めることは現実的ではなかった．そこで，FastMNMF の基盤となる，音源信号から多チャネル混合音が生成される生成モデルに対して，混合音から音源信号を推定する深層分離モデルを導入することで，償却型変分推論 (AVI) の枠組みを用いて，両者を一挙に同時学習する方法を考案した．

2.7 ノンパラメトリックベイズの導入 [9]

理論的には，MNMF は音源数がマイク数よりも多い劣決定条件でも適用できる一方で，FastMNMF は高々マイク数と同じ音源数まで扱うことができる．実用上は，決定条件を前提とした BSS 手法である IVA や ILRMA などと同様に，音源数をマイク数と同一に設定することも多いが，適切に音源数を指定することで，一つの音源が複数の音源として分割されることを防ぐことができる．そこで，理論上は無限個の音源を考えるガンマ過程に基づくノンパラメトリックベイズモデルを定式化し，観測データに合わせて必要な個数の音源を自動的に推定する手法を考案した．

3 スマートグラスを用いたリアルタイム音声強調・認識

我々のグループでは，音響信号処理技術のキラーアプリケーションとして，リアルタイム音声コミュニケーション支援のため，スマートグラスを用いたオンライン音声強調・認識に取り組んでいる（図 2）．この応用においては，事前学習が困難な未知の実環境にどのように適応するのかが重要な課題となる．また，ユーザの身体や頭部の動作に伴うマイクアレイの位置変化（相対的音源方向の変化）も問題となる．これらは，研究室環境下での従来のオフラインベンチマークでは存在しなかった極めて困難な課題である．

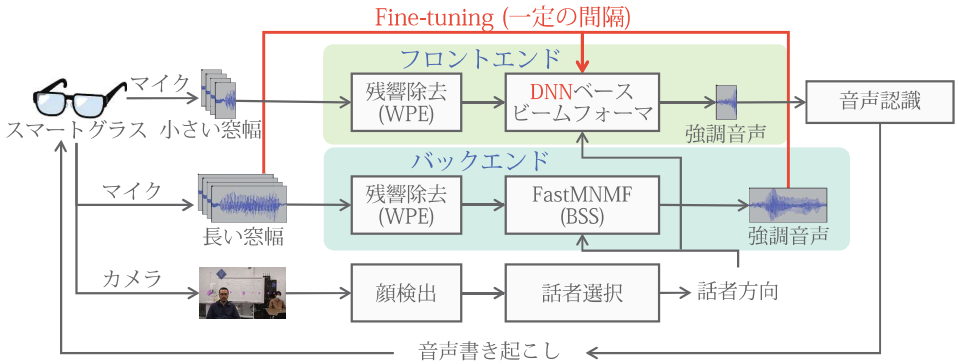


図 2: スマートグラスを用いたリアルタイム音声強調・認識とオンライン適応

スマートグラスを用いた音声強調において，FastMNMF は，適応的なパラメータ推定（教師なし学習）によって未知の環境でも頑健に動作する点で有望であるが，その計算量は依然として大きく，リアルタイム動作は容易ではなかった．一方，多チャネル音声強調の主流である最小分散 (MVDR) ビームフォーミングは，極めて高速に動作する一方で，いかに適切な音声と雑音の空間共分散行列を与えるが性能を左右する．その典型的な解決法の一つは，深層ニューラルネット (DNN) を用いて，観測音（混合音）のスペクトログラムに対して音声マスクを推定するものである．この方式では，音声と雑音の空間共分散行列は適応的に計算されるが，DNN は事前に教師あり学習する必要があるため，学習時と運用時の環境のミスマッチによって性能が大幅に低下する問題は避けられない．

このような性質に鑑み，我々は，FastMNMF をバックエンド，ビームフォーミングをフロントエンドとした音声強調部を構築し，音声認識部と一体的に運用することを試みている．具体的には，バックエンドでは，十分な反復推定が行える一定の間隔で，大きなサイズの時間窓に対して FastMNMF を行うことで，音声と雑音の空間共分散行列を推定することができる．これらに基づく最小分散 (MVDR) ビームフォーミングを行えば音声強調を行うことができる [11]．DNN ベースのビームフォーミングでは，観測音（混合音）と FastMNMF で分離された音声とのペアデータを用いて音声マスク推定用の DNN を Fine-tuning することで環境へのオンライン適応が実現できる [12]．

我々はさらに，マスク推定用の DNN と音声認識用の DNN とを結合し，運用時に全体を Fine-tuning することにも取り組んでいる [13]．この方式は，音声強調と音声認識との同時最適化に関する研究に着想を得ているが，運用時には音声認識の正解データが存在しない点が本質的に異なる．そのため，認識結果から，出力の長さが十分にあり，音声認識モデルおよび大規模言語モデルが与えるスコアがいずれも大きいものを，擬似的な正解データとして選択する方法を考案している．

4 おわりに

本稿では，実環境下での音声コミュニケーションを支援するため，特定の話者の音声をリアルタイムで強調・認識・翻訳・拡張現実 (AR) 表示を行うスマートグラスの開発について紹介した．音響信号処理分野では，深層学習の導入により，音源分離や音声強調などの基盤技術で目覚ましい進展があったが，キラーアプリケーションが不在であった．近い将来，スマートグラスはスマートコンタクトレンズまで小型化されることは間違いなく，我々は，スマートフォンに変わる，人の視聴覚と密接にリンクしたウェアラブルデバイスへの応用こそがキラーアプリケーションになると確信している．実際，Meta や Google などから同様の目的をもったスマートグラスの開発プロジェクトがつい最近相次いで発表されており，この研究課題はにわかに注目を浴びつつある．我々は昨年からいち早く取り組み，すでにいくつかの成果発表を行ってきているが，さらに研究を加速させていきたい．

謝辞

本稿で紹介した一連の研究は，京都大学 大学院情報学研究科 知能情報学専攻 音声メディア分野および理化学研究所 革新知能統合研究センター (AIP) 音響情景理解チームにて実施したものである．中でも，關口航平氏（当時理研 AIP/京大大学院生，現在 DENSO），Aditya Arie Nugraha 氏（理研 AIP），Mathieu Fontaine 氏（当時理研 AIP，現在 Télécom Paris），坂東宜昭氏（産総研），Yicheng Du 氏（当時京大大学院生）らの貢献は特に大きく，彼らの協力なくしては成し得なかった．その他，多くの研究者・学生との議論を通じて，数多くの有益な助言を得た．この場を借りて最大限の謝意を表明したい．

また，本研究の一部は，JSPS 科研費 No. 20H00602, 20K21813, 19H04137 および JST さきがけ No. JPMJPR20CB の支援を受けた．

参考文献

[1] Kouhei Sekiguchi, Aditya A. Nugraha, Yoshiaki Bando, and Kazuyoshi Yoshii. Fast multichannel source separation based on jointly diagonalizable spatial covariance matrices. In *European Signal Processing Conf. (EUSIPCO)*, pp. 1–5, 2019.

[2] Kouhei Sekiguchi, Yoshiaki Bando, Aditya A. Nugraha, Kazuyoshi Yoshii, and Tatsuya Kawahara. Fast multichannel nonnegative matrix factorization with directivity-aware jointly-diagonalizable spatial covariance matrices for blind source separation. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 28, pp. 2610–2625, 2020.

[3] Kouhei Sekiguchi, Yoshiaki Bando, Aditya A. Nugraha, Kazuyoshi Yoshii, and Tatsuya Kawahara. Semi-supervised multichannel speech enhancement with a deep speech prior. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 27, No. 12, pp. 2197–2212, 2019.

[4] Aditya A. Nugraha, Kouhei Sekiguchi, Mathieu Fontaine, Yoshiaki Bando, and Kazuyoshi Yoshii. Flow-based independent vector analysis for blind source separation. *IEEE Signal Processing Letters*, Vol. 27, pp. 2173–2177, 2020.

[5] Yoshiaki Bando, Kouhei Sekiguchi, Yoshiki Masuyama, Aditya A. Nugraha, Mathieu Fontaine, and Kazuyoshi Yoshii. Neural full-rank spatial covariance analysis for blind source separation. *IEEE Signal Processing Letters*, Vol. 28, pp. 1670–1674, 2021.

[6] Aditya A. Nugraha, Kouhei Sekiguchi, Mathieu Fontaine, Yoshiaki Bando, and Kazuyoshi Yoshii. Flow-based fast multichannel nonnegative matrix factorization for blind source separation. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 501–505, 2022.

[7] Kouhei Sekiguchi, Yoshiaki Bando, Aditya A. Nugraha, Mathieu Fontaine, and Kazuyoshi Yoshii. Autoregressive fast multichannel nonnegative matrix factorization for joint blind source separation and dereverberation. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 511–515, 2021.

[8] Kouhei Sekiguchi, Yoshiaki Bando, Aditya A. Nugraha, Mathieu Fontaine, Kazuyoshi Yoshii, and Tatsuya Kawahara. Autoregressive moving average jointly-diagonalizable spatial covariance analysis for joint source separation and dereverberation. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 30, pp. 2368–2382, 2022.

[9] Yoshiaki Bando, Kouhei Sekiguchi, and Kazuyoshi Yoshii. Gamma process FastMNMF for separating an unknown number of sound sources. In *European Signal Processing Conf. (EUSIPCO)*, pp. 291–295, 2021.

[10] Mathieu Fontaine, Kouhei Sekiguchi, Aditya A. Nugraha, Yoshiaki Bando, and Kazuyoshi Yoshii. Generalized fast multichannel nonnegative matrix factorization based on gaussian scale mixtures for blind source separation. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 30, pp. 1734–1748, 2022.

[11] Aditya A. Nugraha, Kouhei Sekiguchi, Mathieu Fontaine, Yoshiaki Bando, and Kazuyoshi Yoshii. DNN-free low-latency adaptive speech enhancement based on frame-online beamforming powered by block-online FastMNMF. In *IEEE Int. Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 1–5, 2022.

[12] Kouhei Sekiguchi, Aditya A. Nugraha, Yicheng Du, Yoshiaki Bando, Mathieu Fontaine, and Kazuyoshi Yoshii. Direction-aware adaptive online neural speech enhancement with an augmented reality headset in real noisy conversational environments. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1–8, 2022.

[13] Yicheng Du, Aditya A. Nugraha, Kouhei Sekiguchi, Yoshiaki Bando, Mathieu Fontaine, and Kazuyoshi Yoshii. Direction-aware joint adaptation of neural speech enhancement and recognition in real multiparty conversational environments. In *Annual Conf. of Int. Speech Communication Association (Interspeech)*, pp. 2918–2922, 2022.

[14] Hiroshi Sawada, Hirokazu Kameoka, Shoko Araki, and Naonori Ueda. Multichannel extensions of non-negative matrix factorization with complex-valued data. *IEEE Trans. on Audio,*

Speech, and Language Processing, Vol. 21, No. 5, pp. 971–982, 2013.

[15] Daichi Kitamura, Nobutaka Ono, Hiroshi Sawada, Hirokazu Kameoka, and Hiroshi Saruwatari. Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 24, No. 9, pp. 1626–1641, 2016.

[16] Ngoc Q. K. Duong, Emmanuel Vincent, and Remi Gribonval. Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Trans. on Audio, Speech, and Language Processing*, Vol. 18, No. 7, pp. 1830–1840, 2010.

[17] Kazuyoshi Yoshii. Correlated tensor factorization for audio source separation. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 731–735, 2018.

[18] Kazuyoshi Yoshii, Koichi Kitamura, Yoshiaki Bando, Eita Nakamura, and Tatsuya Kawahara. Independent low-rank tensor analysis for audio source separation. In *European Signal Processing Conf. (EUSIPCO)*, pp. 1657–1661, 2018.

[19] Nobutaka Ito and Tomohiro Nakatani. FastMNMF: Joint diagonalization based accelerated algorithms for multichannel nonnegative matrix factorization. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 371–375, 2019.

[20] Nobutaka Ito, Rintaro Ikeshita, Hiroshi Sawada, and Tomohiro Nakatani. A joint diagonalization based efficient approach to underdetermined blind audio source separation using the multichannel wiener filter. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 29, pp. 1950–1965, 2021.

[21] Nobutaka Ito and Tomohiro Nakatani. FastFCA-AS: Joint diagonalization based acceleration of full-rank spatial covariance analysis for separating any number of sources. In *IEEE Int. Workshop on Acoustic Signal Enhancement (IWAENC)*, pp. 151–155, 2018.

[22] Nobutaka Ito and Tomohiro Nakatani. Multiplicative updates and joint diagonalization based acceleration for under-determined bss using a full-rank spatial covariance model. In *IEEE Global Conf. on Signal and Information Processing (GlobalSIP)*, pp. 231–235, 2018.

[23] Nobutaka Ono. Stable and fast update rules for independent vector analysis based on auxiliary function technique. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 189–192, 2011.

[24] Aditya A. Nugraha, Kouhei Sekiguchi, and Kazuyoshi Yoshii. A flow-based deep latent variable model for speech spectrogram modeling and enhancement. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, Vol. 28, pp. 1104–1117, 2020.

[25] Paris Smaragdis. Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, Vol. 22, No. 1, pp. 21–34, 1998.

[26] Taesu Kim, Torbjørn Eltoft, and Te-Won Lee. Independent vector analysis: An extension of ICA to multivariate components. In *Int. Conf. on Independent Component Analysis and Signal Separation (ICA)*, pp. 165–172, 2006.

[27] Keigo Kamo, Yuki Kubo, Norihiro Takamune, Daichi Kitamura, Hiroshi Saruwatari, Yu Takahashi, and Kazunobu Kondo. Joint-diagonalizability-constrained multichannel nonnegative matrix factorization based on multivariate complex Student's t-distribution. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA ASC)*, pp. 869–874, 2020.

[28] Keigo Kamo, Yuki Kubo, Norihiro Takamune, Daichi Kitamura, Hiroshi Saruwatari, Yu Takahashi, and Kazunobu Kondo. Joint-diagonalizability-constrained multichannel nonnegative matrix factorization based on multivariate complex sub-Gaussian distribution. In *European Signal Processing Conf. (EUSIPCO)*, pp. 890–894, 2021.

[29] Mathieu Fontaine, Kouhei Sekiguchi, Aditya A. Nugraha, and Kazuyoshi Yoshii. Unsupervised robust speech enhancement based on alpha-stable fast multichannel nonnegative matrix factorization. In *Annual Conf. of Int. Speech Communication Association (Interspeech)*, pp. 4541–4545, 2020.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-06

ドローン聴覚におけるヒストグラム情報と周波数情報を用いた音源定位性能向上の検討

Improvement of sound source localization performance in drone audition using histogram and frequency information

小松崎和泉 [1*]　干場功太郎 [1]　岩附信行 [1]
Izumi Komatsuzaki[1]　Kotaro Hoshiba[1]　Nobuyuki Iwatsuki[1]

[1] 東京工業大学
[1] Tokyo Institute of Technology

Abstract: ドローンを用いた被災者捜索のための音源探査技術において，これまで，著しい時刻変化を伴うドローン自身のエゴノイズに対する頑健性，広い捜索範囲，低い計算コスト，汎用性をすべて満たす音源定位手法の開発を目的に，ヒストグラム情報を用いて空間スペクトルにおけるエゴノイズの除去を動的に行う音源定位手法の提案を行った．本論文では，これまでの提案手法における，エゴノイズと目標音の方向が近い場合に定位性能が低下するといった問題点を解決するために，周波数方向の情報も利用して空間スペクトルを三次元的に解析することにより，エゴノイズと近い方向に存在する目標音に対しても定位が可能となるよう提案手法の改良を行った．実環境での屋外実験とシミュレーションにより，提案手法の性能を評価した結果，本論文で紹介する改良した提案手法を用いることでノイズ付近の目標音も定位可能となり，高いノイズ耐性と広い捜索可能範囲の両者を同時に満足することができ，本手法の有用性が確認された．

1　はじめに

近年，災害地において，人が侵入できない場所にも容易に侵入できること，迅速な活動ができることから，ドローンを用いた要救助者の捜索手法が注目されている．ドローンを用いた捜索では，カメラによる方法が一般的であるが [1]，暗い時間帯の捜索活動，および瓦礫等に埋もれた被災者といったカメラに映らない対象の捜索は困難である．そこで，音情報による捜索手法の確立を目的に，ドローン搭載マイクロホンアレイを用いた音源探査技術の研究が行われている [2]．ドローンを用いた音源探査の実用化にあたり，課題の一つがドローンのエゴノイズである．風や飛行状態の影響により時刻変化を伴うエゴノイズに対する頑健性，広い捜索範囲，ドローン搭載の小型コンピュータを用いてリアルタイムで処理を行うための低い計算コスト，どのような機体・状況でも捜索可能な汎用性を持った音源探査手法が求められる．

これまで，音源探査手法として，MUSIC (MUltiple SIgnal Classification) 法 [3] に基づく音源定位手法が提案されてきている．音源定位において多く用いられる，一般的な MUSIC 法である SEVD-MUSIC (MUSIC based on Standard Eigen Value Decomposition) 法は，計算コストが低い反面，ノイズ耐性が低い．そこで，事前収録したノイズの相関行列を用いてノイズ成分を除去する GEVD-MUSIC (MUSIC based on Generalized Eigen Value Decomposition)[4] や GSVD-MUSIC (MUSIC based on Generalized Singular Value Decomposition)[5] が提案された．これらは，SNR (Signal-to-Noise Ratio) の低い状況でも高い音源定位性能を持つが，計算コストが高く，事前に収録した過去のノイズ情報を用いているため汎用性がない．さらに，時刻変化するノイズへの耐性の強化および汎用性を補うことを目的に，直前の時刻の収録音をノイズと仮定して処理を行う iGEVD-MUSIC (incremental GEVD-MUSIC)[6] や iGSVD-MUSIC (incremental GSVD-MUSIC)[7] が提案されている．しかし，これらは，計算コストが高いことに加え，著しく時刻変化するノイズに対する耐性は不十分である．また，Hoshiba らは，一定の方向からエゴノイズが到来するようなマイクロホンアレイを設計し，SEVD-MUSIC にて得られた空間スペクトルに対し，目標音の捜索範囲からエゴノイズの到来範囲を事前に除外することでノイズ耐性を向上させる手法を提案した [8]．しかし，除外範囲をあらかじめ設定する必要があり，狭い範囲を除外すると時刻変化を伴うノイズに対する十分な耐性が見込めず，広い範囲を除外すると捜索範囲が狭まる．

これらの問題を解決するため，これまでに，過去の情報を用いず，得られた現在の空間スペクトルから，ヒストグラム情報に基づきノイズの判定を行い，動的に捜索範囲の制限を行う手法を提案した [9]．しかし，ノイズと目標音の方向が近い場合に，目標音がノイズと誤判定されてしまうことから，捜索可能な目標音方向が限られるといった問題点があった．

*連絡先：東京工業大学　工学院　機械系
〒 152-8552　東京都目黒区大岡山 2-12-1 I1-26
E-mail: komatsuzaki.i.aa@m.titech.ac.jp

本稿では，著しい時刻変化を伴うノイズに対する頑健性，広い捜索範囲，低い計算コスト，汎用性をすべて満たす音源定位手法の開発を目的に，提案手法 [9] の改良を行い，より高い定位性能を持つ空間スペクトルにおけるノイズ判定手法を提案する．本手法では，得られた現在の空間スペクトルから，ヒストグラム情報と周波数情報に基づき動的にノイズ判定を行い，目標音成分の抽出を行う．通常，方位角，仰角の二次元で行われる空間スペクトルにおける解析を，周波数情報を含めた三次元行うことで，ノイズ付近の目標音がノイズと誤判定される問題点が解決され，より多くの目標音方向に対して，捜索が可能になると期待される．本稿では基礎検討として，提案手法のノイズ耐性および捜索範囲について，屋外実験及びシミュレーションにより評価した．

2　これまでの提案手法

過去のノイズ情報やモデルを使わず，現在の空間スペクトル情報のみからノイズの判定を行い，動的に探索範囲の制限を行う手法を提案した [9]．ここでは空間スペクトルとして，計算コストが低い SEVD-MUSIC により得られる MUSIC スペクトルを扱う．また，一定の方向からエゴノイズが到来するようなマイクロホンアレイを使用することを想定する．以下にこれまでの提案手法のアルゴリズムを示す．

ある時刻にて得られた空間スペクトル $P(\psi)$ より，ヒストグラム H を算出する．

$$H(p) = \mathrm{histogram}(P(\psi)) \tag{1}$$

ここで，ψ はマイクロホンアレイに対する方位角 θ，仰角 ϕ から，$\psi = (\theta, \phi)$ と定義する．また，p は空間スペクトルのパワーに対する階級である．

一般的に空間スペクトルから音源定位を行う場合，ピーク検出を行うが，目標音成分の最大パワーより大きいエゴノイズ成分がある場合，正確な目標音源の定位ができない．そこで，目標音成分の最大パワーより小さい基準値を定義し，基準値より大きいエゴノイズ成分を除去することで，ピーク検出により正しい定位が可能になる．また，目標音成分の最大パワー近傍の基準値を取ることで，目標音の定位に影響のあるエゴノイズ範囲のみが除外でき，捜索範囲が最大になると考えられる．本稿では，空間スペクトル全要素のヒストグラムのピーク以降の変曲点が，目標音成分の最大パワーより小さいかつ近傍という傾向があったため，試験的に基準値をそのように定める．H の二階微分を求め，変曲点における空間スペクトルのパワー p_t を得る．

$$H''(p) = \frac{d^2 H}{dp^2} \tag{2}$$

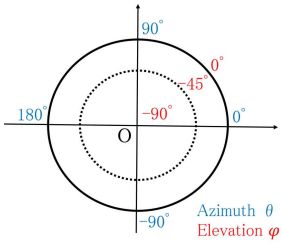
$$p_t = p|_{H''(p)=0} \tag{3}$$

エゴノイズは一定方向より到来することから，基準方向 $\psi_0 = (\theta_0, \phi_0)$ を設定し，ψ_0 を含む，p_t 以上のパワーを持つ連続した方向 Ψ をノイズとみなす．

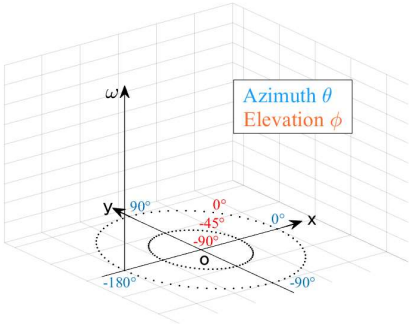
$$\Psi = \{\psi|_{P(\psi)>p_t}\} \ni \psi_0 \tag{4}$$

得られた Ψ を除外した範囲を捜索範囲とし，捜索範囲にて $P(\psi)$ が最大値をとる方向を目標音の方向 ψ_{target} として検出する．

$$\psi_{target} = \mathrm{argmax}_{\psi \notin \Psi}(\mathrm{P}(\psi)) \tag{5}$$



(a) 二次元 MUSIC スペクトルの座標設定.



(b) 三次元 MUSIC スペクトルの座標設定.

図 1: 座標設定.

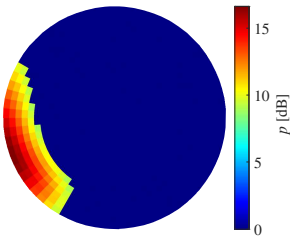
3　提案手法の改良

前章の手法を用いた場合，エゴノイズと目標音の方向が近い場合，目標音の成分がエゴノイズが存在する範囲と重なることが原因で，目標音がエゴノイズと誤判定されるといった問題点があった．その問題点を解決するため，周波数情報を取り入れ，提案手法の改良を行う．これまで二次元で解析していた空間スペクトルを，周波数情報を含めた三次元で解析することで，エゴノイズと目標音の方向が近い場合であっても，それらの周波数特性の違いから分離が可能になると期待できる．以下に提案手法のアルゴリズムを示す．

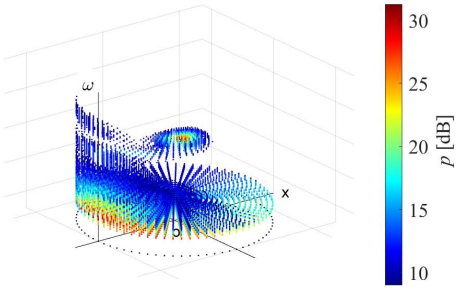
SEVD-MUSIC 法では，周波数毎に算出した空間スペクトル $P_\omega(\psi)$ を周波数方向に加算した，空間スペクトル $P(\psi)$ を解析に用いる．

$$P(\psi) = \sum_{\omega} P_\omega(\psi) \tag{6}$$

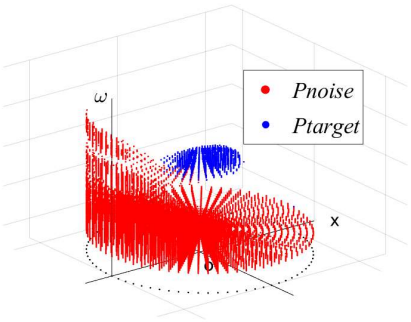
ここで，ω とは周波数ビンを表す．一方，本手法では，加算する前の空間スペクトル $P_\omega(\psi)$ に着目する．ある



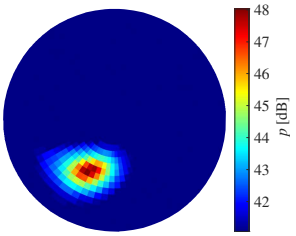
(a) $Pcut_\omega$.



(b) $Pcut$.



(c) $Pcut$ より分離されたエゴノイズ成分 $Pnoise$（赤色），および目標音成分 $Ptarget$（青色）.



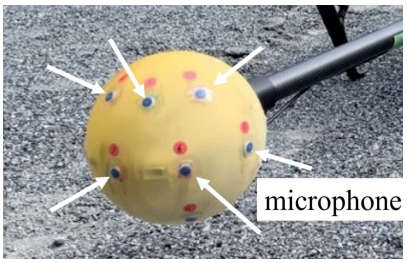
(d) P'.

図 2: 改良した提案手法における目標音成分抽出の過程.

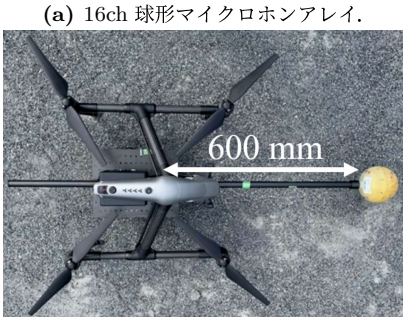
1 つの周波数ビンの空間スペクトル $P_\omega(\psi)$ に対して，p_t 以上のパワーを持つ範囲 $Pcut_\omega(\psi)$ を抽出する.

$$Pcut_\omega(\psi) = P_\omega(\psi)| > p_t \tag{7}$$

得られた $Pcut_\omega(\psi)$ の一例をプロットしたものを図 2a に示す．ここでは，図 1a の設定軸に従ってプロットされており，各方向から到来した音のパワーをカラーマップで示している．紺色の範囲を除く範囲が p_t 以上の範囲である．全周波数ビンにて算出した $Pcut_\omega(\psi)$ の集合を



(a) 16ch 球形マイクロホンアレイ.



(b) マイクロホンアレイ搭載ドローン.

図 3: 実験にて使用したマイクロホンアレイおよびドローン.

表 1: 比較条件

条件 1. SEVD-MUSIC
条件 2. 従来手法 (捜索範囲:−90°〜90°)
条件 3. 従来手法 (捜索範囲:−135°〜135°)
条件 4. 提案手法
条件 5. 提案手法（改）

$Pcut(\omega, \psi)$ とおく．この $Pcut(\omega, \psi)$ の一例をプロットしたものを図 2b に示す．図 2b は図 1b の設定軸に従ってプロットされている．エゴノイズと目標音は，その周波数特性の違いから，このような三次元データとして見ると，エゴノイズと三次元的に連続していない目標音成分があると思われる．そこで，$Pcut(\omega, \psi)$ に対して，ψ_0 を含み，三次元的に連続している部分をエゴノイズ成分 $Pnoise(\omega, \psi)$，その他を目標音成分 $Ptarget(\omega, \psi)$ と分離する．それぞれの成分を図示したものが図 2c であり，赤色が $Pnoise(\omega, \psi)$，青色が $Ptarget(\omega, \psi)$ を表す．そして，$Ptarget(\omega, \psi)$ を周波数方向に足し合わせ，二次元の空間スペクトル $P'(\psi)$ を得る.

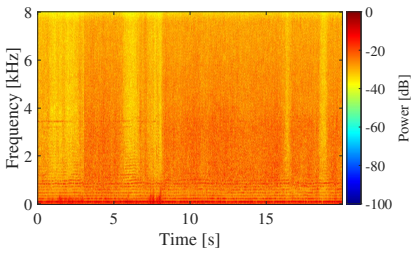
$$P'(\psi) = \sum_\omega Ptarget(\omega, \psi) \tag{8}$$

得られた $P'(\psi)$ の一例をプロットしたものを図 2d に示す．図 2d は図 1a の設定軸に従ってプロットされている．$P'(\psi)$ に対して，最大値をとる方向を目標音方向 $\psi target$ として検出する.

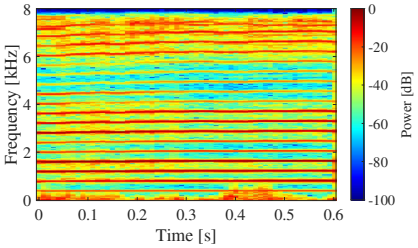
$$\psi_{target} = \mathrm{argmax}_\psi P'(\psi) \tag{9}$$

4 評価実験

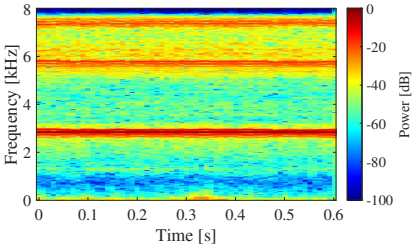
提案手法の性能を検証するため，評価実験を行った．屋外実環境にて収録したエゴノイズと，シミュレーショ



(a) エゴノイズ.



(b) 声.



(c) ホイッスル.

図 4: 評価に使用した音源のスペクトログラム.

ンにより作成した任意の方向から到来した目標音を加算することにより評価用信号を作成し，解析を行った．

エゴノイズは，DJI 社製 Inspire 2 に搭載したマイクロホンアレイにて収録した．マイクロホンアレイには，下半球に 12ch，上半球に 4ch の MEMS マイクロホンが設置されている 16ch 球形マイクロホンアレイ (図 3a)[10] を用いた．本マイクロホンアレイでは，サンプリング周波数 16 kHz，量子化ビット数 24 bit で音響信号が収録される．マイクロホンアレイはドローンの中心から 600 mm の位置に設置した (図 3b)．高度 10 m でホバリング中，および速度 1 m/s，2 m/s，3 m/s で飛行時のエゴノイズを収録した．目標音のサンプルには声およびホイッスルを用い，方位角が $-180° \leq \theta < 180°$，仰角が $-90° \leq \phi \leq 0°$ の範囲で 5° 刻みで到来方向を設定し，幾何計算により得た伝達関数から各方向から到来した目標音を作成した．そして，収録したノイズと作成した目標音を，SNR が 4 dB 刻みで $-20 \sim 0$ dB となるよう加算し，評価用信号を作成した．エゴノイズ，目標音のスペクトログラムを図 4 に示す．

まず，評価用信号を SEVD-MUSIC により解析し，MUSIC スペクトルを求めた．SEVD-MUSIC のパラメータは，目標音源数を 2，相関行列に用いる平均化フレーム数を 50，最小解析周波数を 500 Hz，最大解析周波数を 4000 Hz とした．得られた MUSIC スペクトルに対し，従来手法として，エゴノイズが到来する方向を捜索範囲から一定範囲除外する Hoshiba らの手法 [8]，および提案手法 [9]，本稿で改良した提案手法（以降，提案手法（改）と呼ぶ）により処理を行い比較した．表 1 に比較条件を示す．従来手法の捜索範囲は $-90°$〜$90°$ と $-135°$〜$135°$ の二通りとした．また，提案手法および提案手法（改）における基準方向は，$\psi_0 = (-180°, 0°)$ とした．

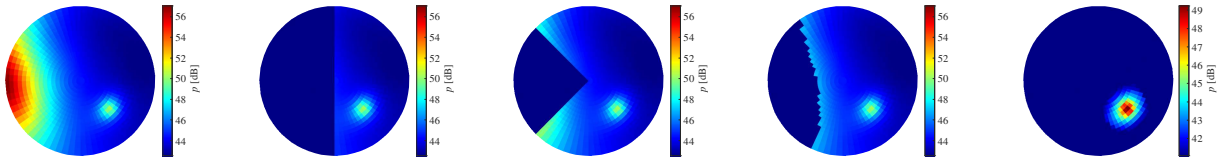
5 結果

様々な条件で作成した評価用信号を，表 1 で示した各手法で処理した結果を図 5 に示す．図 5a〜5e が SNR が -12dB，目標音源がホイッスル，目標音源方向が $\psi_{target} = (-45°, -45°)$ の場合の結果，図 5f〜5j が SNR が -12dB，目標音源がホイッスル，$\psi_{target} = (-180°, -45°)$ の場合の結果，図 5k〜5o が SNR が -12dB，目標音源がホイッスル，$\psi_{target} = (-150°, -10°)$ の場合の結果である．また，5a．5f，5k が条件 1，5b．5g，5l が条件 2，5c．5h，5m が条件 3，5d．5i，5n が条件 4，5e．5j，5o が条件 5 で解析を行った．これらの空間スペクトルは図 1a の設定軸に従ってプロットされている．

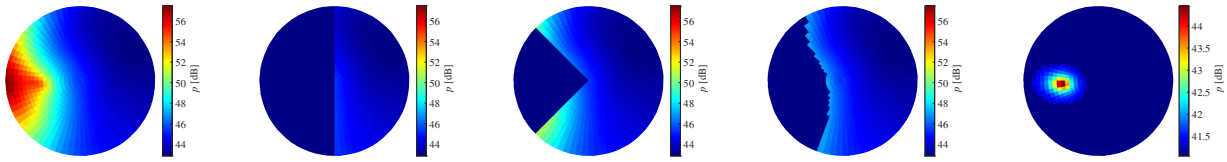
図 5a，5f，5k は SEVD-MUSIC により算出された MUSIC スペクトルであり (条件 1)，左側にエゴノイズが存在することが確認できる．なお，目標音は図 5a では右側，図 5f，5k ではエゴノイズと重なって存在しているが，いずれもエゴノイズの方がパワーが大きく，ピーク検出による目標音方向の定位ができない．図 5b，5c，5g，5h，5l，5m は，図 5a，5f，5k から従来手法によりエゴノイズを除外した結果であり (条件 2，3)，図中における紺色の部分が除外範囲である．条件 2 において，除外範囲は全体の 50%を占めており，捜索可能範囲が大幅に減少していることがわかる．一方，条件 3 は，除外範囲が狭く，捜索可能範囲が広い．また，5d，5i，5n は，提案手法によりエゴノイズを除外した結果 (条件 4) である．条件 2〜4 において，エゴノイズと目標音が離れている結果 1 では空間スペクトルに目標音成分が確認できる．一方で，エゴノイズと目標音が近い結果 2 および結果 3 では，目標音が除外範囲に含まれていることがわかる．図 5e，5j，5o は，図 5a，5f，5k から提案手法（改）によりエゴノイズを除外した結果である (条件 5)．提案手法（改）を用いたところ，結果 1，2 ともに目標音が抽出できている．しかし，目標音源方向がエゴノイズが及ぶ方向と完全に一致する結果 3 では，目標音が抽出できていない．以上の結果から，提案手法（改）は，従来手法およびこれまでの提案手法と比べて，エゴノイズに近い目標音も定位可能であるため捜索範囲が広く，高いノイズへの耐性および広い捜索範囲の両者を満足すると期待できる．

6 考察

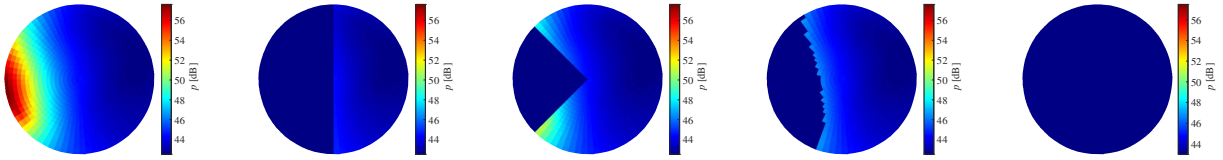
ノイズ耐性と捜索範囲について評価するため，評価用音響信号 80 フレーム (40 秒分)，目標音 2 種類，音



(a) 結果 1: 条件 1. (b) 結果 1: 条件 2. (c) 結果 1: 条件 3. (d) 結果 1: 条件 4. (e) 結果 1: 条件 5.

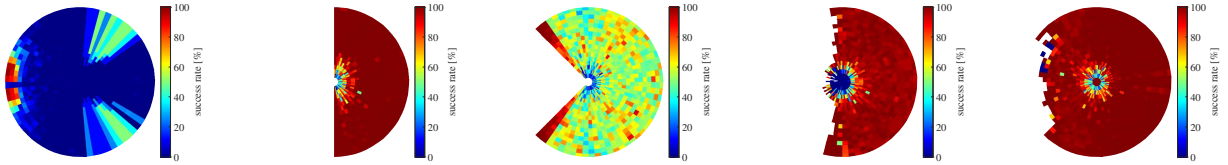


(f) 結果 2: 条件 1. (g) 結果 2: 条件 2. (h) 結果 2: 条件 3. (i) 結果 2: 条件 4. (j) 結果 2: 条件 5.



(k) 結果 3: 条件 1. (l) 結果 3: 条件 2. (m) 結果 3: 条件 3. (n) 結果 3: 条件 4. (o) 結果 3: 条件 5.

図 5: 各手法で得られた空間スペクトル.
結果 1 (SNR: −12 dB，目標音源: ホイッスル，目標音源方向: $\psi_{target} = (-45°, -45°)$)
結果 2 (SNR: −12 dB，目標音源: ホイッスル，目標音源方向: $\psi_{target} = (-180°, -45°)$)
結果 3 (SNR: −12 dB，目標音源: ホイッスル，目標音源方向: $\psi_{target} = (-150°, -10°)$)



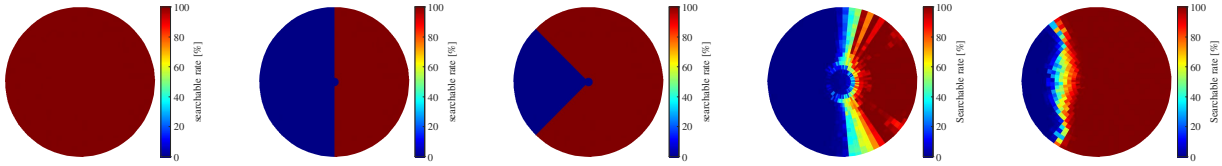
(a) 条件 1. (b) 条件 2. (c) 条件 3. (d) 条件 4. (e) 条件 5.

図 6: 目標音の設定方向毎の定位成功率 (SNR=−12 dB).

源方向 1,297 通り，SNR6 通りの全 1,245,120 回の試行を行った．

初めに，定位成功率について考察する．図 6 に各条件における目標音の設定方向毎の定位成功率を示す．図 1a の設定軸に従いプロットされており，各方向毎の定位成功率をカラーマップで示している．定位成功率は，ψ_{target} が目標音の設定方向 ±5° 以内のとき定位成功とし，成功数 / 目標音が除外範囲に含まれなかった試行数 のように算出した．SEVD-MUSIC である条件 1 は，多くの目標音の方向において，定位成功率が大幅に低下している．また，従来手法にて狭い範囲を除外する条件 3（図 6c）においても，定位成功率が低くなった．一方，従来手法にて広い範囲を除外する条件 2（図 6b），提案手法である条件 4（図 6d）および提案手法（改）である条件 5（図 6e）は，目標音が除外範囲に含まれなかった試行において，高い定位成功率を獲得した．図 8 に全目標音の設定方向における定位成功率を示す．横軸は SNR，縦軸は全方向で平均した定位成功率であり，各条件ごとに図 6 にて捜索可能であった方向における定位成功率を平均し算出した．SNR が −20 ～ − 12dB のときは，条件 5 の提案手法（改）が最も高い定位成功率を獲得した．一方，SNR が −12～0dB のとき，提案手法（改）の定位成功率は条件 2，条件 4 に比べて低下した．この原因として，SNR が高くなるにつれヒストグラム情報から算出される基準値の値が大きくなるため，基準値以下に該当するエゴノイズ成分が発生し，エゴノイズ成分の連続性が失われてしまうことが考えられる．そのため，除外されずに残ったエゴノイズ成分の影響から，提案手法（改）において SNR が高い場合の定位成功率が低下すると考察できる．

次に捜索可能範囲について考察する．図 7 に各条件における目標音の設定方向毎の捜索可能割合を示す．図 1a の設定軸に従いプロットされており，各方向毎の捜索可能割合をカラーマップで示している．捜索可能割



(a) 条件 1.　(b) 条件 2.　(c) 条件 3.　(d) 条件 4.　(e) 条件 5.

図 7: 目標音の設定方向毎の捜索可能割合 (SNR=−12 dB).

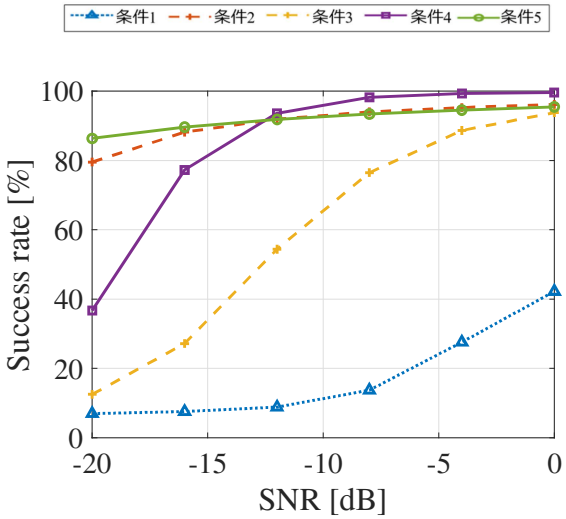


図 8: 全目標音の設定方向における定位成功率.

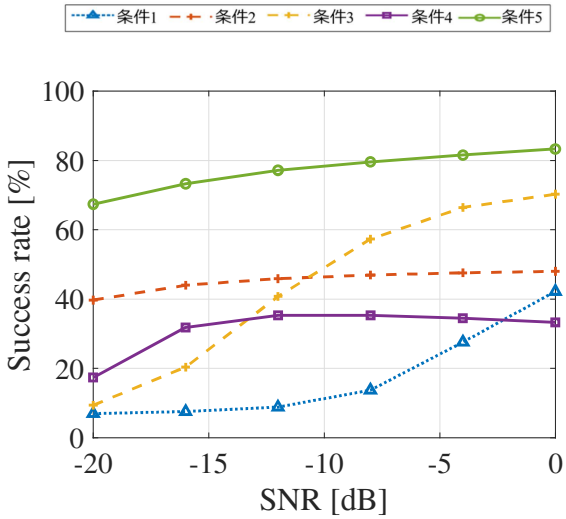


図 10: 捜索可能範囲を考慮した定位成功率.

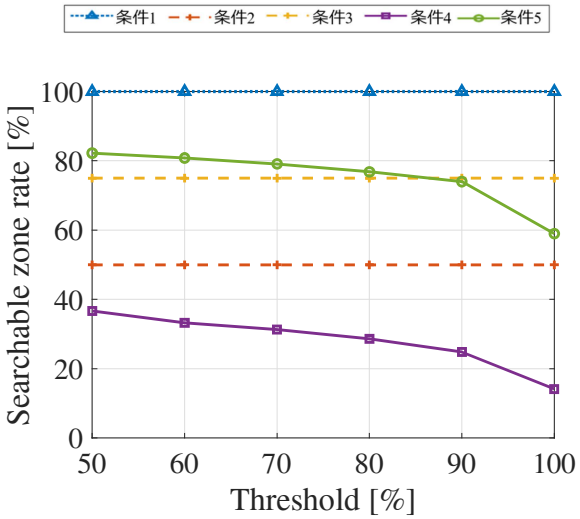


図 9: 全目標音の設定方向における捜索可能範囲.

合は，目標音が除外範囲に含まれなかった試行数 / 試行数　のように算出した．条件 2（図 7b），条件 4（図 7d）において，捜索可能割合が高い目標音方向は全体の半分以下であり，除外範囲に含まれる目標音方向が多い．一方，条件 3（図 7c），条件 5（図 7e）は，多くの目標音方向で捜索可能割合が高い．図 9 に全目標音の設定方向における捜索可能範囲を示す．横軸は閾値，縦軸は捜索可能範囲であり，捜索可能範囲は，図 7 において閾値以上の捜索可能割合を持つ目標音方向 / 全目標音方向　として算出した．条件 5 である提案手法（改）は閾値が 90%以下であるとき，条件 1 を除いた手法の中で，最も高い広い捜索可能範囲を獲得した．条件 4 である提案手法は，前報告 [9] では，捜索可能範囲を，目標音が除外範囲に含まれたか否かを考慮せずに，全解析範囲 − 除外範囲 / 全解析範囲 として算出したため，十分広い捜索可能範囲が得られた．しかし，捜索可能範囲の算出方法を変更した本報告では，すべての条件の中で捜索可能範囲が最小となった．

以上の結果を踏まえ，定位成功率（図 8）と捜索可能範囲（図 9）を統合させた，捜索可能範囲を考慮した定位成功率を図 10 に示す．横軸は SNR，縦軸は捜索可能率である．捜索可能率は，成功数 / 試行数 のように算出し，定位成功率および捜索可能範囲をどちらも考慮した評価値とした．これより，提案手法（改）は従来手法およびこれまでの提案手法に比べて，捜索可能率が大幅に向上していることがわかる．

以上から，提案手法（改）は著しい時刻変化を伴うノイズへの耐性と広い捜索範囲を満足し，実環境におけるドローン搭載マイクロホンアレイを用いた音源探査において有用性があることがわかった．提案手法（改）を用いることにより，前章における図 5i，5j，のように，これまでの提案手法にて問題であった，エゴノイズ付近の目標音がエゴノイズと誤判定され定位が不可

能であるといった問題が解決できた．しかし，図 5n, 5o のように，目標音方向がエゴノイズが及ぶ方向と完全に一致する場合，目標音成分がエゴノイズと判定され除外されてしまい，定位が失敗するという問題も新たに判明した．今後は，このような状況を解決する手法の検討に加え，本稿で評価していない計算コストや汎用性について，様々な条件や他の手法で比較，評価を行っていく予定である．

7 結言

本稿では，著しい時刻変化を伴うノイズに対する頑健性，広い捜索範囲，低い計算コスト，汎用性をすべて満たす音源定位手法の開発を目的に，過去の情報を用いず，得られた現在の空間スペクトルから，ヒストグラム情報と周波数情報に基づきノイズの判定を行い，目標音成分の抽出を行う手法を提案した．これまでの提案手法では，エゴノイズ付近の目標音がエゴノイズと誤判定され定位が不可能であるといった問題点があったが，以前は方向情報のみの二次元で解析していた空間スペクトルを，周波数情報を含めた三次元で解析し提案手法を改良することにより，これまでの提案手法における問題点を解決することができた．評価実験の結果，改良した提案手法により，従来手法およびこれまでの提案手法では同時に満たすことのできなかったノイズ耐性と広い捜索範囲の両者を同時に満足することができ，有用性が確認できた．しかし，目標音の到来方向がエゴノイズが及ぶ方向と完全に一致しているとき，改良した提案手法を用いた場合においても，目標音成分がエゴノイズ判定され除外され，定位が失敗するという問題も新たに判明した．今後は，このような状況を解決する手法の検討に加え，計算コストや汎用性について，様々な条件や他の手法で比較，評価を行っていく．

謝辞

本研究の一部は，JSPS 科研費 22K14218, 公益信託小野音響学研究助成基金および東京工業大学 工学院助教インセンティブ研究経費の助成を受けたものである．

参考文献

[1] 加藤，寺島，高見: 要救助者の複数ドローンによる協調探索のためのエッジサーバ集約型自動スケジューリング手法とシミュレーション評価マルチメディア，分散協調とモバイルシンポジウム *2019* 論文集, pp.291–296 (2019)

[2] K. Hoshiba, O. Sugiyama, A. Nagamine, R. Kojima, M. Kumon, K. Nakadai: Design and Assessment of Sound Source Localization System with a UAV-Embedded Microphone Array *J. of Robotics and Mechatronics* VOL. 29, NO. 1, pp. 154-167 (2017)

[3] R. O. Schmidt: Multiple Emitter Location and Signal Parameter Estimation *IEEE Trans. Antennas and Propagation*,VOL. 34, NO. 3, pp. 276-280 (1986)

[4] K. Nakamura, K. Nakadai, F. Asano, Y. Hasegawa, H. Tsujino: Intelligent Sound Source Localization for Dynamic Environment, *IEEE/RSJ International Conference on Intelligent Robots and Systemd* (2009)

[5] K. Nakamura, K. Nakadai, G. Ince: Real-time Super-resolution Sound Source Localization for Robots *Proc. of IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS)* pp. 694-699 (2012)

[6] K. Okutani, T. Yoshida, K. Nakamura, K. Nakadai: Intelligent Sound Source Localization for Dynamic Environment, *Outdoor Auditory Scene Analysis Using a Moving Microphone Array Embedded in a Quadrocopter* (2012)

[7] T. Ohata, K. Nakamura, A. Nagamine, T. Mizumoto, T. Ishizaki, R. Kojima, O. Sugiyama, K. Nakadai: Outdoor Sound Source Detection Using a Quadcopter with Micrphone Array *J. of Robotics and Mechatronics* VOL. 29, NO. 1, pp. 177-187, (2017)

[8] K. Hoshiba, K. Washizaki, M. Wakabayashi, T. Ishiki, M. Kumon, Y. Bando, D. Gabriel, K. Nakadai, H. G. Okuno: Design of UAV-Embedded Microphone Array System for Sound Sorce Localization in Outdoor Environments *Sensors* VOL. 17, NO. 11, pp. 1-16, (2017)

[9] 小松崎，干場，武田，菅原: ヒストグラム情報を用いた時刻変化の著しい雑音に対する体制の高い音源定位手法の提案日本ロボット学会 (2022)

[10] K. Nonami, K. Hoshiba, K. Nakadai, M. Kumon, H.G. Okuno, Y. Tanabe, K. Yonezawa, H. Tokutake, S. Suzuki, K. Yamaguchi, S. Sunada, T. Takaki, T. Nakata, R. Noda, H. Liu, S. Tadokoro: Recent R&D Technologies and Future Prospective of Flying Robot in Tough Robotics Challenge *Disaster Robotics - Results from the ImPACT Tough Robotics Challenge, Satoshi Tadokoro Ed., Springer International Publishing* pp. 77–142, (2019)

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-07

複数音源追跡におけるドローン群の行動計画の検討

Study of drone swarm action planning in multiple sound source tracking

山田 泰基[1*]　糸山 克寿[1,2]　西田 健次[1]　中臺 一博[1]
Taiki Yamada[1], Katsutoshi Itoyama[1], Kenji Nishida[1], Kazuhiro Nakadai[1,2]

[1] 東京工業大学
[1] Tokyo Institute of Technology
[2] (株) ホンダ・リサーチ・インスティチュート・ジャパン
[2] Honda Research Institute Japan Co.,Ltd.

Abstract: 本稿はマイクロホンアレイ搭載ドローン群による複数音源追跡を行うため，ドローン配置およびマイクロホンアレイ配置の最適化手法を提案する．一般的にマイクロホンアレイは，信号処理を通じて音源方向を推定できるツールとして用いられている．マイクロホンアレイと搭載することで，ドローンは災害現場などで助けを呼ぶ人の捜索が可能になり，マイクロホンアレイ搭載ドローンが複数台あればその位置を推定・追跡できるようになる．しかし，マイクロホンアレイ搭載ドローン群による音源位置追跡の性能はマイクロホンアレイの配置の影響が大きく，なるべく各マイクロホンアレイの推定方向同士は直交し，かつマイクロホンアレイ-音源間の距離は小さい方が望ましい．そこで，本稿では音源位置追跡のためのマイクロホンアレイの配置最適化手法を提案し，その有効性を検討する．提案手法は数値シミュレーションを通じて検証され，提案手法がマイクロホンアレイの推定方向同士を直交に近づけながらも音源との距離大きく離さないはたらきが見られた．シミュレーション上ではおよそ 10 m 離れた音源 2 つに対してそれぞれ 2.15 m, 0.65 m の RMSE (Root Mean Square Error) で追跡できたことを確認した．

1　はじめに

近年，災害現場におけるドローンの活用が期待されている．特にドローン群の制御技術が向上したことにより，災害フィールドにドローン群を配備し短時間でフィールドの環境理解を進められることが期待される．ドローン聴覚の文脈では，マイクロホンアレイと呼ばれる聴覚センサを搭載することで音源方向を推定し，瓦礫に埋もれて視認しづらい要救助者を発見する研究が報告されている [1–3]．以前に我々もマイクロホンアレイを搭載したドローン群による移動音源位置追跡について報告した [4,5]．複数マイクロホンアレイを用いることで同タイミングの方向推定・三角測量が可能になり，推定したい音源軌道への収束が早いというメリットがあるという手法である．しかし，既存研究ではドローンの位置に関する考察が少なく，実際にはドローンの配置によって三角測量の精度が落ちたり，ドローンが音源から遠すぎてしまい音源がほとんど聞き取れなかったりすることがあるため，移動音源追跡の精度向上にドローン配置の最適化は不可欠である．そこで，我々は以前に単独移動音源追跡のためのドローン配置およびマイクロホンアレイ配置の最適化手法を提案した．本稿では，本手法を複数音源の追跡に拡張し，数値シミュレーションを通じてその有効性を検討する．複数音源追跡が単独音源追跡と異なる点は，全てのマイクロホンアレイが全ての音源の追跡に寄与できない点である．一般的に単独音源の位置追跡を行う際は，各マイクロホンアレイは音源を囲むように移動することで，音源から離れ過ぎず，ドローンノイズとの SN 比を大きくした状態で位置推定を行うことができる．しかし，複数音源を追跡する際，音源同士が離れている場合は全てのマイクロホンアレイが各音源を聞き取れる位置に置くことは困難である．そのため，各音源の追跡に用いるマイクロホンアレイの情報は取捨選択する必要があり，聞き取れるマイクロホンアレイだけで音源追跡とマイクロホンアレイ配置の最適化を考える必要がある．そこで本稿では，各マイクロホンアレイが各音源の方向を推定できたかという「確信度」なる変数を定義し，この概念を元に音源追跡に用いるマイ

*連絡先：東京工業大学
〒 152-8552 東京都目黒区大岡山 2-12-1 西 8 号館 W-30
E-mail: yamada@ra.sc.e.titech.ac.jp

クロホンアレイを選択し，その配置を最適化する．

2 提案手法

複数マイクロホンアレイを用いて音源位置追跡を行う場合，マイクロホンアレイの配置が追跡精度に影響を及ぼす可能性がある [6]. 例えば，複数マイクロホンアレイで音源方向推定し，三角測量的に音源位置を推定するとき，マイクロホンアレイ同士の方向が音源から見て近しいと，その方向に音源位置推定誤差が乗りやすくなる場合がある．また，マイクロホンアレイをドローンに搭載する場合，常にドローンノイズがマイクロホンアレイに近い箇所で発生するため，ドローンが音源から離れるほど，追跡音源との SN 比が著しく小さくなる恐れがある．そこで本稿ではマイクロホンアレイを搭載したドローン群を用いた複数音源追跡の性能を向上するべく，ドローン群の配置を最適化するアルゴリズムを提案する．

2.1 提案アルゴリズム概要

音源追跡を行うにあたって，マイクロホンアレイ搭載ドローンが N 台，追跡音源が S 個存在するシナリオを考える．以後，$i, j \in 1, \ldots, N$ はドローンに搭載されたマイクロホンアレイのインデックス，$k \in 1, \ldots, S$ は音源のインデックスを指す．提案手法では各ドローンは Algorithm 1 に従い移動することで音源位置追跡の向上を図る．Algorithm 1 を文章化した内容は以下の通りである．

0. 変数の初期化は 2.5 節を参照
1. MUSIC 法を用いて音源方向推定を行う．
2. MUSIC スペクトルに応じてマイクロホンアレイ i の音源 k に対する確信度 $p(\alpha_{i\to k}|\boldsymbol{z})$ を式 (1) より計算する．確信度に応じて各音源の位置推定にどのマイクロホンアレイが寄与するかを決定する．
3. 推定方向と選択したマイクロホンアレイ群を元に各音源に対応するマイクロホンアレイ群毎に音源位置追跡を行う．
4. 推定した音源位置を元に式 (5) を通じて最適なマイクロホンアレイ配置を計算し，各ドローンは算出された位置に移動する．

続く小節は上記の行程の詳細を説明する．

Algorithm 1 提案アルゴリズム (1 タイムステップ分)

Require: $\hat{\boldsymbol{x}}_{i,t}$
for $i = 1, \ldots, N$ **do**
  $\boldsymbol{d}_i, P(\phi, \theta) \leftarrow$ MUSIC 法 [7] による推定方向と MUSIC スペクトル
end for
for $k = 1, \ldots, K$ **do**
  $\mathcal{M}_k \leftarrow \emptyset$
  for $i = 1, \ldots, N$ **do**
    $p(\alpha_{i\to k}|\boldsymbol{z}) \leftarrow$ 式 (1)
    if $p(\alpha_{i\to k}|\boldsymbol{z}) \geq p_{\text{thre}}$ **then**
      $\mathcal{M}_k$ に i を追加
    end if
  end for
  $\hat{\boldsymbol{s}}_{k,t} \leftarrow \mathcal{M}_k$ に含まれるマイクロホンアレイを用いた MT-GSFT [4] による推定位置
end for
$\boldsymbol{x}_{t+1} \leftarrow$ 式 (5)

2.2 音源方向推定およびマイクロホンアレイの確信度の更新

一般に複数マイクロホンアレイを用いて音源位置を推定する場合，複数マイクロホンアレイで音源方向推定を行い，三角測量的に音源位置を推定することが多い．本稿では音源方向推定の手段として，MUSIC 法 [7] と呼ばれる手法を用いて音源位置推定を行う．MUSIC 法は収録された多チャネル音響信号の空間相関行列 $\boldsymbol{R}$ が張る固有空間を解析する手法であり，目的音源の部分空間と雑音部分空間の直交性を用いて音源の方位・仰角を推定する手法である．MUSIC 法では MUSIC スペクトルと呼ばれる空間スペクトル $P(\phi, \theta)$ を算出し，スペクトル P のピーク値に位置する方位角 ϕ と仰角 θ を推定方向とする．しかし，広いフィールドで音源位置追跡を行う場合，遠距離にある音源は減衰してしまい全てのマイクロホンアレイが遠距離音源の方向推定を行えるとは限らない．そのため音源位置追跡を行う際，音源を聞き取れたマイクロホンアレイのみで行うことが望ましい．本稿ではマイクロホンアレイの「確信度」を定義し，音源 k に対して確信度が高いマイクロホンアレイの方向推定結果のみを用いて音源位置追跡を行う．マイクロホンアレイ i が音源 k を観測するイベントを $\alpha_{i\to k}$ と定義し，$\alpha_{i\to k} = 1$ で観測が成功，$\alpha_{i\to k} = 0$ で観測が失敗したと定義する．このとき，マイクロホンアレイ i の音源 k に対する確信度 $p(\alpha_{i\to k}|\boldsymbol{z})$ は式 (1) で更新される確率で定義される．

$$p(\alpha_{i\to k}|\boldsymbol{z}) = \frac{p(\alpha_{i\to k})p(\boldsymbol{z}|\alpha_{i\to k})}{\sum p(\alpha_{i\to k})p(\boldsymbol{z}|\alpha_{i\to k})} \tag{1}$$

ここで，$\boldsymbol{z}$ は観測を表し，確信度 $p(\alpha_{i\to k}|\boldsymbol{z})$ は観測 $\boldsymbol{z}$ が与えられたときマイクロホンアレイ i が音源 k を観測できている確率である．事前確率 $p(\alpha_{i\to k})$ は前タイムステップの事後確率によって与えられ，尤度 $p(\boldsymbol{z}|\alpha_{i\to k})$ は式 (2), (3) で計算する．

$$p(\boldsymbol{z}|\alpha_{i\to k}) = \begin{cases} P_{\mathrm{norm}}(\phi_{i\to k}, \theta_{i\to k}) & (\alpha_{i\to k} = 1) \quad (2) \\ \dfrac{1}{N_\phi N_\theta} & (\alpha_{i\to k} = 0) \quad (3) \end{cases}$$

$P_{\mathrm{norm}}(\phi, \theta)$ は MUSIC スペクトルの総和が 1 になるように正規化した MUSIC スペクトルであり，N_ϕ, N_θ はそれぞれ MUSIC スペクトルを計算するときの方位角ビン，仰角ビンの数である．また，$(\phi_{i\to k}, \theta_{i\to k})$ はマイクロホンアレイ i から見た音源 k の方位角，仰角であるため，$P_{\mathrm{norm}}(\phi_{i\to k}, \theta_{i\to k})$ はマイクロホンアレイ i から見た音源 k の正規化 MUSIC スペクトルにあたる．一般に正しい方向推定が行える場合，MUSIC スペクトルは音源方向にピークを立てるため，$P_{\mathrm{norm}}(\phi, \theta)$ は $\alpha_{i\to k} = 1$ のときの尤度分布と見なし式 (2) のように定義した．また，音源 k が正しく観測できない場合の MUSIC スペクトルの一般的な分布の形は存在しないため，$\alpha_{i\to k} = 0$ である場合の尤度は MUSIC スペクトルが全方向において同値である場合の値と定義した．よって，マイクロホンアレイ i から見た音源 k の確信度は，マイクロホンアレイ i から見た音源 k の方向にあたる MUSIC スペクトルがピークに近い値を取るときに増加して 1 に近づき，そうでない場合は減少して 0 に近づく．本アルゴリズムでは確信度 $p(\alpha_{i\to k}|\boldsymbol{z})$ は全マイクロホンアレイ・音源の組み合わせに対して計算し，この確信度に基づいて音源 k の位置追跡に用いるマイクロホンアレイの取捨選択を行う．具体的には，音源 k に対するマイクロホンアレイ i の確信度 $p(\alpha_{i\to k}|\boldsymbol{x})$ が閾値 p_{thre} 以上である場合，そのマイクロホンアレイは集合 $\mathcal{M}_k \subseteq \{1, \ldots, N\}$ に加えられ，続く音源位置追跡とマイクロホンアレイ配置の最適化に用いられる．

2.3 音源位置追跡

前小節で求めたマイクロホンアレイ集合 $\mathcal{M}_k$ を用いて，音源位置追跡を行う．本稿では，以前我々が提案した音源位置追跡は MT-GSFT 法 [4] を用いて追跡を行う．MT-GSFT はドローンに搭載された複数マイクロホンアレイによる音源位置追跡のために提案された手法で，ドローンノイズで音源方向推定が大きく分散するような状況下でも外れ値となるような三角測量点の影響を抑えるはたらきを持つ．具体的には複数マイクロホンアレイの推定方向による三角測量によって得られた三角測量点群を $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_Q\}$ と置き，これらの三角測量点を式 (4) のような混合ガウス分布に置き換える．

$$\sum_{q=1}^{Q} \frac{1}{Q} \cdot \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}) \tag{4}$$

ただし，$\mathcal{N}$ はガウス分布を表し，$\boldsymbol{\Sigma}$ はヒューリスティックに決める分散定数である．式 (4) で得られた混合ガウス分布を，Gaussian Sum Filter (GSF) に適用することで追跡したい音源の位置分布を得られ，混合ガウス分布の重み付き平均を取ることで推定音源位置 $\hat{\boldsymbol{s}}$ が得られる．注意すべき点は，音源 k について MT-GSFT 法による追跡を行いたい場合は，マイクロホンアレイ集合 $\mathcal{M}_k$ に含まれたマイクロホンアレイの方向推定結果のみを用いて三角測量点の計算を行う点である．

2.4 マイクロホンアレイ配置の最適化

本稿では (i) 各マイクロホンアレイ推定方向同士が直交に近づき，かつ (ii) ドローンが音源に近い状況が，音源位置追跡に有効であるという仮定のもと，式 (5) を構成し，本式を最小化することで最適なマイクロホンアレイ配置を算出する．

$$\underset{\boldsymbol{x}_{t+1}}{\mathrm{argmin}}\, f(\boldsymbol{x}_{t+1}) + \lambda_g g(\boldsymbol{x}_{t+1}) + \lambda_h h(\boldsymbol{x}_{t+1}) \tag{5}$$

$$\mathrm{s.t.} z_i \geq z_{\mathrm{lim}} \tag{6}$$

$$f(\boldsymbol{x}_{t+1}) = \sum_{k=1}^{S} \sum_{\{i,j\}\in\mathcal{M}_k \ (i\neq j)} \frac{(\boldsymbol{x}_{i,t+1} - \hat{\boldsymbol{s}}_{k,t})^\top (\boldsymbol{x}_{j,t+1} - \hat{\boldsymbol{s}}_{k,t})}{\|\boldsymbol{x}_{i,t+1} - \hat{\boldsymbol{s}}_{k,t}\|_2 \|\boldsymbol{x}_{j,t+1} - \hat{\boldsymbol{s}}_{k,t}\|_2} \tag{7}$$

$$g(\boldsymbol{x}_{t+1}) = \sum_{k=1}^{S} \sum_{i\in\mathcal{M}_k} \|\boldsymbol{x}_{i,t+1} - \hat{\boldsymbol{s}}_{k,t}\|_2 \tag{8}$$

$$h(\boldsymbol{x}_{t+1}) = \sum_{i=1}^{N} \|\boldsymbol{x}_{i,t+1} - \boldsymbol{x}_{i,t}\|_2 \tag{9}$$

ここで，$\boldsymbol{x}_{i,t} \in \mathbb{R}^3$ は時刻 t におけるマイクロホンアレイ i の 3 次元位置であり，$\boldsymbol{x}_t = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ は全マイクロホンアレイの時刻 t における三次元位置の集合である．z_i はマイクロホンアレイ i の z 座標，z_{lim} は z 座標の下限であり，式 (6) はドローンが音源に近づきすぎないための制約である．また，$\boldsymbol{s}_{k,t} \in \mathbb{R}^3$ は時刻 t における音源 k の 3 次元位置である．$f(\boldsymbol{x})$ は 2 つのマイクロホンアレイが正しい方向を推定した場合の 2 本の方向ベクトルの角度の余弦の総和である．よって，理想的な推定方向同士が直交化すると $f(\boldsymbol{x})$ は 0 に近づくため，$f(\boldsymbol{x})$ の最小化によって推定方向同士が直交に近づき三角測量点の分散の抑制が期待できる．$g(\boldsymbol{x})$ は音源とマイクロホンアレイの距離の総和であり，$g(\boldsymbol{x})$ を最小化することによって音源とドローンの距離を抑えることができ，距離減衰による SN 比の減少を抑えることができる．$h(\boldsymbol{x})$ は前タイムステップのドローン位

置と現在のタイムステップのドローン位置の距離を最小化することで，ドローンが大きく移動することを防ぐはたらきを持つ．λ_g, λ_h はそれぞれ $g(\boldsymbol{x}), h(\boldsymbol{x})$ にかかる係数である．$f(\boldsymbol{x})$ の値はは大きくても高々$S \cdot {}_N C_2$ までしか取らない一方，$g(\boldsymbol{x}), h(\boldsymbol{x})$ はドローンと音源の距離やドローンの移動距離の総和を取るため，一般的には $f(\boldsymbol{x})$ は $g(\boldsymbol{x}), h(\boldsymbol{x})$ よりも著しく小さくなることが多い．そこで，項同士のバランスを取るために係数 λ_g, λ_h を設けており，本稿の評価シミュレーションでは $\lambda_g = 0.01, \lambda_h = 0.0001$ と設定した．また，本稿ではマイクロホンアレイはドローンから突き出ている形で搭載されていることを想定しており，そのためマイクロホンアレイの後方でプロペラがドローンノイズを発することになる．よって音源はドローンの後方ではなく，ドローンの前方にあることが好ましい．本アルゴリズムではドローンの姿勢は推定音源位置 $\hat{\boldsymbol{s}}_k$ を向くように配置し，マイクロホンアレイ i が複数音源に対して確信度が閾値 p_{thre} を超える場合は該当する音源群への平均方向を向くように配置した．

2.5　初期タイムステップの処理

上記のアルゴリズムを実行する際，音源数は既知である必要があり，また確信度 $p(\alpha_{i\to k})$ の初期値が必要になる．そこで，本稿では音源数は S は既知と仮定し，確信度 $p(\alpha_{i\to k})$ の初期値は全ての音源・マイクロホンアレイのペアに置いて 0.5 と設定する．また，MT-GSFT を行う上で初期事前分布を設定する必要があるが，本稿では全マイクロホンアレイから算出した三角測量点を k-means 法で S 個のクラスターに分割し，各クラスターの重心を平均，$\boldsymbol{\Sigma}_{\text{gen}} \in \mathbb{R}^3$ を分散とした正規乱数生成器から I 個の 3 次元の点を作り出し，それぞれの点を平均に持つ分散 $\boldsymbol{\Sigma}_0$ の正規分布を合成した混合ガウス分布を初期事前分布に用いる．

3　評価シミュレーション

提案手法によるマイクロホンアレイ配置の最適化によって複数音源の位置追跡が可能になるかを確かめるため，数値シミュレーションを行い，その有効性を検証した．

3.1　シミュレーション設定

提案手法の有効性を評価するため，提案手法を MATLAB 上で実装した．本シミュレーションでは音源 1 と音源 2 の 2 個の音源を配置し，6 台のドローンでそれぞれの音源位置を追跡することを考える．(図 1) 以下図 1 で円形の軌跡を描く音源を音源 1，四角形の軌跡

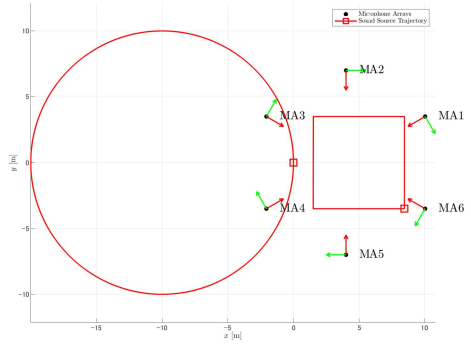


図 1: 上から見たフィールド図．各音源は赤四角マーカーから等速で反時計回りに移動する．赤矢印・緑矢印は各マイクロホンアレイの初期姿勢を示す．(MA= マイクロホンアレイ)

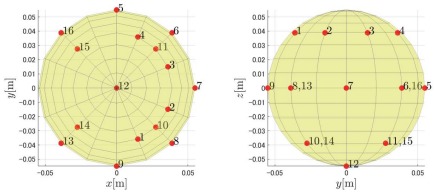


図 2: 球形 16 ch マイクロホンアレイを構成するマイクロホン配置

を描く音源を音源 2 と記述する．図 1 は上から見た俯瞰図であるが，実際には 3 次元の位置追跡を行っており，各音源は高さ $z = 1.5$ m に位置する．また，各ドローンの初期位置は $z = 5$ 平面で点 $(x, y, z) = (4, 0, 5)$ を中心に半径 7 m の円弧上に等間隔に置き，両音源を囲むように配置した．各ドローンには 16ch の球形マイクロホンアレイ (図 2) を 1 台ずつ搭載しており，24bit，16 kHz で収録を行う．収録は $T = 46$ 秒間行い，追跡対象である音源は T 秒間かけて図の赤線の軌道をちょうど 1 周するように進む．音源 1 と音源 2 はそれぞれ 1000 Hz, 2000 Hz の正弦波を絶えず出力している．また，本シミュレーション環境を実際の屋外環境と近づけるため，各マイクロホンアレイには予め収録された 16 ch のドローンノイズを付加しており，音源信号との SN 比は −35 dB と設定した．各ドローンは式 (5) の更新則に従い移動し，更新に必要なパラメータは $\lambda_g = 0.01, \lambda_h = 0.0001, z_{\text{lim}} = 4.79$ と設定した．式 (5) の最小化の計算には内点法を用いた．音源方向推定には MUSIC 法 [7] を適用し，$\omega_L = 900$ Hz から $\omega_H = 2100$ Hz の間の信号より方向推定に用いる空間スペクトルを方位角・仰角ともに 5 度刻みで算出する．

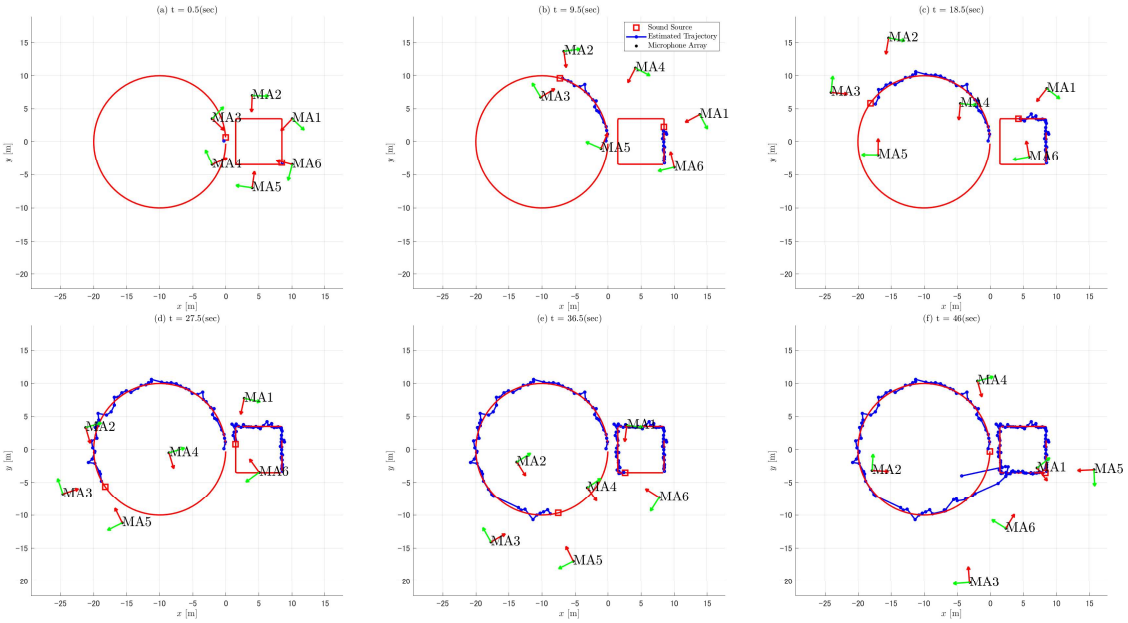


図 3: 音源位置追跡結果

3.2 結果・考察

図 3 は 46 秒間の収録の間に得られた音源位置の追跡結果とマイクロホンアレイ配置であり，提案手法は両音源の軌跡の概形を捉えていることが分かる．また，シミュレーション中盤では MA2, 3, 4, 5 が音源 1 を等間隔で囲んでおり，MA1, 4, 6 が音源 2 を等間隔で囲むように移動している様子が見られた．これは式 (5) を通じた最適化によって，各マイクロホンアレイの推定方向同士が直交に近づこうとするからであると思われる．また，各マイクロホンアレイの高さは常時 $z_{\mathrm{lim}} = 4.79$ m であり，かつ観測できる音源とはおよそ 10 m 前後の距離を保とうする動きが見られており，式 (5) の最適化を通じて各マイクロホンアレイが推定方向同士の直交性を守りつつも音源になるべく近づこうとしている様子が窺える．音源 1（円軌道）の RMSE (Root Mean Square Error) は 2.15 m, 音源 2（四角軌道）の RMSE は 0.65 m であった．音源 1 の RMSE が比較的大きいのは，図 3(f) で見られるように，追跡結果が音源 2 の軌道に引っ張られてしまったからである．本稿では追跡するフィルター同士が共通する音源の位置を推定してしまったとき，同じ音源を追跡しないようにする処置を施していないため，図 3(f) のように近い音源軌道に追跡結果がドリフトしてしまう現象が起きたと考えられる．実際に，ドローン群に別の初期配置を与えてシミュレーションしたとき，2 つの音源追跡フィルターが同じ音源を終始追跡してしまい，6 台のドローンが音源 2 だけを囲むケースが見られた．よって，音源同士が近寄った場合に追跡フィルターが音源を混同しないような処置の必要性が窺えた．

図 4 は各マイクロホンアレイの音源に対する確信度 $p(\alpha_{i\to k})$ の推移を示したものである．例えば，初期位置が音源 1 に近く，音源 2 から遠かった MA2, 3 は音源 1 に対する確信度が増加し，音源 2 に対する確信度が現象したことが見られる．そのため，MA2, 3 は音源 1 の音源追跡に寄与することに集中し，終始音源 1 の周囲を移動していることが分かる．音源 2 の追跡においても同様のことが MA1, 6 について見られる．MA4, 5 については，初期位置は比較的音源 1, 2 の両方に近かったことから，両方の音源の方向を推定できていたことが確認された．よって，シミュレーション序盤の両音源に対する MA4, 5 の確信度は $P_{\mathrm{thre}} = 0.3$ を超えており，両音源の追跡に寄与していた様子が見られる．ただし，式 (5) によって各マイクロホンアレイは推定方向同士が直交するように配置する作用がはたらき，マイクロホンアレイ同士の位置はなるべく離れるように移動するようになる．そのため，MA4 は常に音源 1 と 2 の間に配置し両音源の方向推定に努める一方で，MA5 は音源 1 がある方へ引っ張られ，やがて音源 1 の追跡のみに寄与するようになった．以上より，式 (1) による確信度の更新を通じて，各マイクロホンアレイの確信度は音源を聞き取れる（= 音源に近く SN 比が大きい）場合は上昇し，そうでない場合は減少する意図通りの傾向が見られた．また，提案アルゴリズム

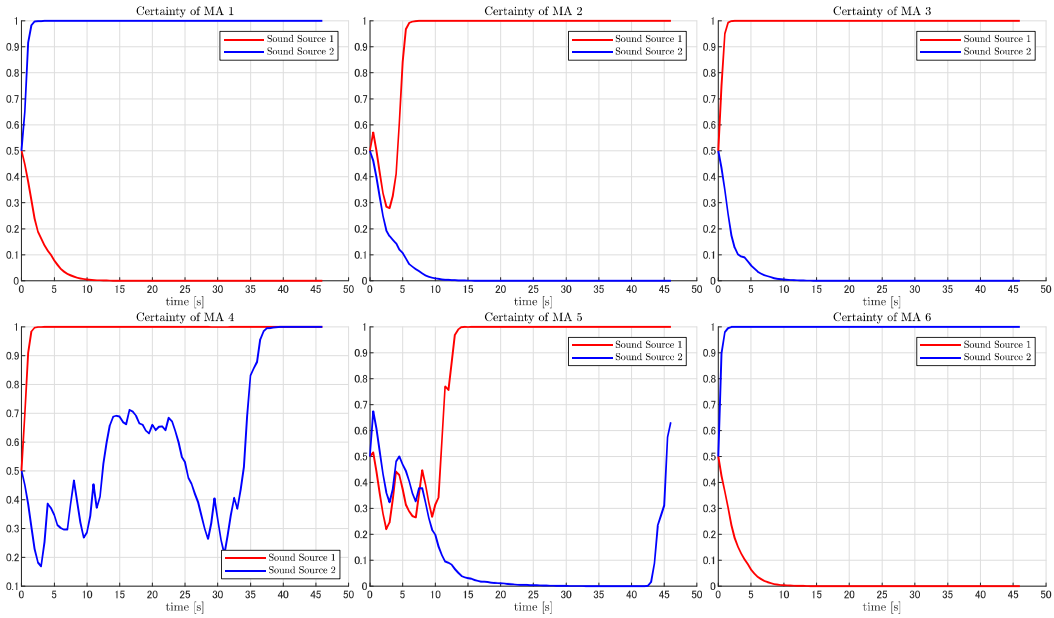


図 4: 各マイクロホンアレイの各音源に対する確信度の推移

によって音源の位置追跡とマイクロホンアレイ配置の更新は，その音源に対する確信度が高いマイクロホンアレイのみによって行われていることが窺える．

4　終わりに

本稿では，マイクロホンアレイ搭載ドローン群による複数音源の追跡のためのマイクロホンアレイ配置の最適化アルゴリズムを提案した．一般的に複数マイクロホンアレイで音源位置を推定する際に，各マイクロホンアレイが推定する音源方向同士は直交であることが好ましく，それと同時に音源が聞き取れるようにマイクロホンアレイは追跡音源に近づくことが望ましい．この 2 条件を満たすために評価関数式 (5) を構築し，ドローンおよびマイクロホンアレイ配置を決定するアルゴリズムを提案した．また，複数の音源を同時に聞き取れないマイクロホンアレイが存在しても推定結果を損なわないように，確信度（式 (1)）という概念を導入し，確信度の高いマイクロホンアレイのみで音源追跡・マイクロホンアレイ配置の最適化が行えるようにした．提案アルゴリズムの有効性を検討すべく数値シミュレーションを行い，提案アルゴリズムが複数音源の追跡を行えるケースが確認できた一方，音源同士が近い時に追跡結果の混同が起こることが確認できた．また，MT-GSFT のような三角測量を行う手法では，1 つの音源につきマイクロホンアレイが少なくとも 2 個以上必要であり，もし追跡途中である音源を聞き取れるマイクロホンアレイが 1 個しかない状況下でも音源位置追跡を継続できるような，追跡戦略のスイッチングが求められる．まとめると，追跡フィルター同士の重なりの解消と音源を追跡できるマイクロホンアレイが 1 台のみになっても追跡が続けられる追跡手法のスイッチングが今後の課題である．加えて，音源の発生と消滅の検知や他の音源追跡手法についても有効であるかの検証も課題の 1 つである．

謝辞

本研究は JSPS 科研費 JP19K12017, JP19KK0260 および JP20H00475 の助成を受けた．

参考文献

[1] K. Nakadai, M. Kumon, H. G. Okuno, K. Hoshiba, M. Wakabayashi, K. Washizaki, T. Ishiki, D. Gabriel, Y. Bando, T. Morito, R. Kojima, and O. Sugiyama, “Development of microphone-array-embedded uav for search and rescue task,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5985–5990.

[2] K. Hoshiba, K. Washizaki, M. Wakabayashi, T. Ishiki, M. Kumon, Y. Bando, D. Gabriel, K. Nakadai, and H. Okuno, "Design of uav-embedded microphone array system for sound source localization in outdoor environments," *Sensors*, vol. 17, no. 11, p. 2535, 2017.

[3] M. Wakabayashi, K. Washizaka, K. Hoshiba, K. Nakadai, H. G. Okuno, and M. Kumon, "Design and implementation of real-time visualization of sound source positions by drone audition," in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 814–819.

[4] T. Yamada, K. Itoyama, K. Nishida, and K. Nakadai, "Sound source tracking by drones with microphone arrays (forthcoming)," in *Proceedings of 2020 IEEE/SICE International Symposium on System Integration: 12-15 January 2020; Honolulu*, 2020.

[5] ——, "Sound source tracking using integrated direction likelihood for drones with microphone arrays," in *2020 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2021, pp. 394–399.

[6] ——, "Assessment of sound source tracking using multiple drones equipped with multiple microphone arrays," *International journal of environmental research and public health*, vol. 18, no. 17, p. 9039, 2021.

[7] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

Rotor noise power spectral density informed sound source enhancement and localisation for unmanned aerial vehicles

Benjamin Yen[*1] Yusuke Hioka[*2]

[*1] Tokyo Institute of Technology [*2] Acoustics Research Centre, University of Auckland

Recent years saw a significant increase in attention to using unmanned aerial vehicles (UAVs) to perform audio signal processing-related applications, such as sound source enhancement and localisation. However, the high levels of UAV rotor noise often result in extremely low input signal-to-noise ratios (SNR), thus rendering the problem highly challenging. This article reviews selected methods shown to perform well in these scenarios. The methods primarily rely on estimating the rotor noise power spectral densities (PSD) by utilising multi-sensory information and or machine learning to achieve the desired accuracy and robustness, thereby creating an effective postfilter or noise masking envelope to reduce the effects of rotor noise.

1. Introduction

Audio source enhancement and localisation have spanned decades of extensive research over many applications. Naturally, with the significant increase in the popularity of unmanned aerial vehicles (UAVs), research for UAV-specific applications has also been gaining increasing attention over the recent decade. For most studies, this involves attaching a microphone array [1] onto the UAV to perform audio recording. However, the proximity of the microphone array to the UAV and the high levels of UAV rotor noise make this problem a uniquely challenging task.

Effectively, the setting results in very low signal-to-noise ratios (SNR), for which many of the classic audio processing techniques are insufficient to be useful. Despite this, recent years saw several studies attempting to perform audio-related applications using UAVs, such as sound source localisation [2, 3, 4], sound source separation [5], and sound source enhancement [6, 7, 8, 9].

For sound source enhancement, a natural solution would involve using a noise mask or a noise filter to reduce the effects of rotor noise. For example, authors in [7] utilised a Kurtosis based noise estimator to design a noise mask, assisted by information regarding the sound source's location. A number of studies also utilised deep neural networks (DNN) to improve source enhancement performance, whether to design a noise mask to use for a postfilter [8] or to optimise beamformer steering [10].

As shown in the studies mentioned earlier, despite the multitude of different approaches, most require a denoising scheme for their methods to be effective. This is also well-demonstrated with the study in [6, 11], utilising the well-known *beamforming with Wiener postfilter* framework [1]. Here, the critical requirement for the method to perform effectively is to have accurate estimates of each sound source's power spectral densities (PSD). The study in [6] realised this using multiple beamformers to design a multi-channel Wiener postfilter (MWF), which was later extended by the studies [9, 12] with a multi-sensory, machine learning based rotor noise PSD estimator. In particular, using non-acoustical features yielded from the rotor's state improves the accuracy and robustness in estimating the rotor noise PSDs. This is demonstrated in [9] with its strong source enhancement performance with real-life experiments using a flying drone, which will be reviewed in this article along with its baseline from [6].

For sound source localisation, a common approach includes the use of the multiple signal classification algorithm (MUSIC), combined with denoising techniques, to achieve the desired performance [2, 3, 4]. Recent studies also showed the use of multiple UAVs to triangulate and improve localisation performance [13]. On the other hand, several studies have also shown the use of the generalised cross-correlation - phase transform (GCC-PHAT) method along with a denoising scheme [14, 15]. This was particularly apparent in the 2019 IEEE Signal Processing Cup, where many of the top participating teams made use of such a framework [16]. The study in [14] in particular demonstrated the effectiveness of using a DNN-driven noise envelope to improve source localisation performance by reducing the effects of rotor noise directly, which has shown to be particularly effective over many input scenarios. This method will also be reviewed in this article.

The rest of this article is organised as follows. First, studies in sound source enhancement for UAVs mentioned earlier will be reviewed in Section 2.. This includes the problem setup, the *beamforming with Wiener postfilter* framework proposed by [6], and extensions introduced in [9]. A sound source localisation algorithm that carries means to reduce rotor noise from [14] will be reviewed in Section 3.. Finally, the article is concluded with some comments and discussions in Section 4..

Contact: Benjamin Yen, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552, E-mail: benjamin@ra.sc.e.titech.ac.jp

Benjamin Yen is an International Research Fellow of the Japan Society for the Promotion of Science.

2. Sound source enhancement

This section reviews methods from studies [6, 9].

2.1 Problem setup

The problem considers a UAV, mounted with an M-sensor microphone array, receiving a target source $S(\omega, t)$, K *spatially coherent* interfering noise sources $N_\theta(\omega, t)$ (including noise generated by U ($\leq K$) UAV rotors) arriving from different angles θ, and ambient *spatially incoherent* noise. The system aims to extract a clear target source signal from the M-channel noisy recordings [6].

The short-time Fourier transform (STFT) of the M-microphone input signals are expressed in vector form as

$$\begin{aligned}\mathbf{x}(\omega, t) &:= \Big[X_1(\omega, t), \quad \cdots, \quad X_M(\omega, t)\Big]^T \\ &= \mathbf{a}_{\theta_0}(\omega) S(\omega, t) + \sum_{u=1}^{U} \mathbf{a}_{\theta_u}(\omega) N_{\theta_u}(\omega, t) \\ &+ \sum_{n=U+1}^{K} \mathbf{a}_{\theta_n}(\omega) N_{\theta_n}(\omega, t) + \mathbf{v}(\omega, t), \end{aligned} \tag{1}$$

where ^{T} denotes the transpose, and $X_m(\omega, t)$ is the STFT of the m-th microphone's input signal. θ_0, θ_u and θ_n indicate the angles to the target, the u-th rotor, and the n-th spatially coherent interfering noise source, respectively. $\omega = 1, \ldots, F$ and t denote the angular frequency (of F frequency bins) and frame index, respectively. $\mathbf{a}_\theta(\omega) = [A_{1,\theta}(\omega), \cdots, A_{M,\theta}(\omega)]^T$ and $\mathbf{v}(\omega, t) = [V_1(\omega, t), \cdots, V_M(\omega, t)]^T$ are the steering vector between the source located at angle θ and each microphone m, and the incoherent noise vector observed by the microphone array, respectively.

In the study in [9], given that both UAV rotor noise and spatially coherent interfering noise can be modelled as spatially coherent sources, $\mathbf{v}(\omega, t)$ is considered negligible for simplicity. In addition, sound sources are assumed to be mutually uncorrelated. For a UAV which typically operates in open outdoor environments, sound propagation is assumed to closely resemble a free field. Regardless, $A_{m,\theta}(\omega)$ is modelled as the transfer function between each sound source and microphone or impulse response (IR) measurements in practice. Furthermore, the problem is assumed to be limited to overdetermined cases, where $M \geq K + 1$. Finally, the problem assumes that the sound arrival angles of the target source and all noise sources are given *a priori*.

2.2 Beamforming with rotor noise informed Wiener postfilter

Figure 1 shows an overview of the source enhancement algorithm from [9]. Beamforming is a commonly used technique to perform source enhancement. In the studies [9, 12], $K + 1$ fixed beamformers are used, with the main lobe of each beamformer directed towards the angle of each sound source θ outlined in Section 2.1 (i.e. θ_0 for the target, θ_u for the u-th rotor noise and θ_n for the n-th interfering noise source). The beamformer outputs $Y_\theta(\omega, t)$ are then calculated as

$$Y_\theta(\omega, t) = \mathbf{w}_\theta^H(\omega) \mathbf{x}(\omega, t), \tag{2}$$

where $\mathbf{w}_\theta(\omega) = [W_{1,\theta}(\omega), \cdots, W_{M,\theta}(\omega)]$ denotes the vector of the beamformer's filter weights and ^{H} denotes the Hermitian conjugate. The framework from [6] used the minimum variance distortionless response (MVDR) beamforming technique, where its weights $\mathbf{w}_\theta$ are obtained as [17]

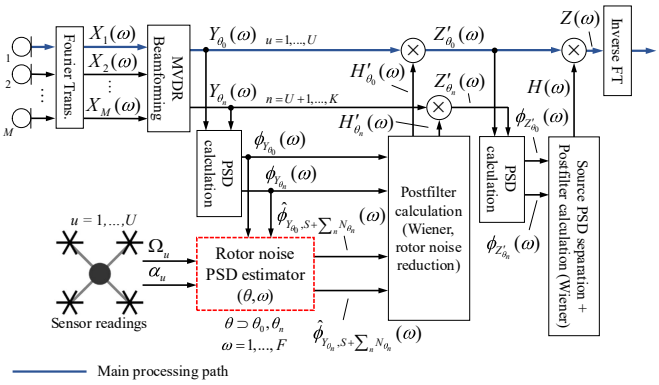


Figure 1: Overall framework of the beamforming with rotor noise informed postfilter method [9].

$$\mathbf{w}_\theta(\omega) = \frac{R^{-1}(\omega)\mathbf{a}_{\theta_{0'}}(\omega)}{\mathbf{a}_{\theta_{0'}}^H(\omega) R^{-1}(\omega) \mathbf{a}_{\theta_{0'}}(\omega)}, \tag{3}$$

where $\mathbf{a}_{\theta_{0'}}(\omega)$ denotes the steering vector of the chosen target source direction for the beamformer (i.e. the angle to which the directivity of the beamformer points). Assuming the measured IRs of each angle of interest is known, $R(\omega)$ is the normalised noise covariance matrix modelled using the steering vector of the N' chosen noise source directions for the beamformer $\mathbf{a}_{\theta_{n'}}(\omega)$ (i.e. the "nulls" of the beamformer).

Often, beamforming is coupled with a postfilter to provide additional reduction of unwanted noise, especially when the number of microphones available is limited [18]. However, as mentioned in Section 1., this framework requires highly accurate estimates of the individual source PSDs to work well. This is particularly challenging due to the very low SNR levels in the UAV problem setting, leading to the target sources and other interfering noise sources being masked heavily by rotor noise. However, rotor noise is strongly correlated to the UAV's state characteristics, which opens a gateway to accurately estimate rotor noise PSDs (i.e. $\phi_{Y_\theta, N_{\theta_u}}(\omega, t)$), which can later be utilised to infer source PSDs of other types. The studies in [19, 20] leveraged this idea and utilised machine learning-based algorithms to estimate $\phi_{Y_\theta, N_{\theta_u}}(\omega, t)$, which was later utilised for source enhancement in [9, 12]. As a result, two Wiener filters (WF) are used. We note that PSDs from microphone signals from studies reviewed in this article are calculated using the Welch method [21].

The first postfilter is dedicated to suppressing rotor noise (hereafter referred to as WF_{rot}). Here, WF_{rot} carries out rotor noise suppression directly on beamformer outputs pointing the target source $Y_{\theta_0}(\omega, t)$ and interfering noise source $Y_{\theta_n}(\omega, t)$, respectively. Here, WF_{rot} is designed using rotor noise PSDs estimated by a rotor noise PSD estimation module. The module estimates the rotor noise PSDs for beamformer outputs that point towards the target source $\sum_{u=1}^{U} \hat{\phi}_{Y_{\theta_0}, N_{\theta_u}}(\omega, t)$ and interfering noise source $\sum_{u=1}^{U} \hat{\phi}_{Y_{\theta_n}, N_{\theta_u}}(\omega, t)$ via a machine learning-based

mapping function, taking the UAV's non-acoustical parameters as its input features (see Section 2.3). Using estimates $\sum_{u=1}^{U} \hat{\phi}_{Y_{\theta_0},N_{\theta_u}}(\omega,t)$ and $\sum_{u=1}^{U} \hat{\phi}_{Y_{\theta_n},N_{\theta_u}}(\omega,t)$, the beamformer output PSDs after removing rotor noise $\hat{\phi}_{Y_{\theta_0},S+\sum_n N_{\theta_n}}(\omega,t)$ and $\hat{\phi}_{Y_{\theta_n},S+\sum_n N_{\theta_n}}(\omega,t)$ are then obtained as

$$\hat{\phi}_{Y_\theta,S+\sum_{n=U+1}^{K} N_{\theta_n}}(\omega,t) = \phi_{Y_\theta} - \sum_{u=1}^{U} \hat{\phi}_{Y_\theta,N_{\theta_u}}. \tag{4}$$

Using these PSDs, $\mathrm{WF}_{\mathrm{rot}}$ is then calculated to reduce rotor noise in $\phi_{Y_{\theta_0}}(\omega,t)$ and $\phi_{Y_{\theta_n}}(\omega,t)$ as

$$H'_\theta(\omega,t) = \frac{\hat{\phi}_{Y_\theta,S+\sum_{n=U+1}^{K} N_{\theta_n}}}{\phi_{Y_\theta}}. \tag{5}$$

Using these Wiener filters, the rotor noise reduced output signals $Z'_{\theta_0}(\omega,t)$ and $Z'_{\theta_n}(\omega,t)$ are then obtained as

$$Z'_\theta(\omega,t) = H'_\theta(\omega,t) Y_\theta(\omega,t). \tag{6}$$

Following (6), a second stage postfiltering process $\mathrm{WF}_{\mathrm{int}}$ is used to further reduce interfering noise, using *PSD estimation in beamspace* [6]. Here, the corresponding PSDs of the multiple beamformers and sound sources are then represented in a set of equations as

$$\underbrace{\begin{bmatrix} \phi_{Z'_{\theta_0}} \\ \phi_{Z'_{\theta_{U+1}}} \\ \vdots \\ \phi_{Z'_{\theta_K}} \end{bmatrix}}_{\mathbf{\Phi}_{Z'_\theta}(\omega,t)} = \underbrace{\begin{bmatrix} |D_{0,\theta_0}|^2 & |D_{0,\theta_{U+1}}|^2 & \cdots & |D_{0,\theta_K}|^2 \\ |D_{U+1,\theta_0}|^2 & |D_{1,\theta_{U+1}}|^2 & \cdots & |D_{U+1,\theta_K}|^2 \\ \vdots & \vdots & \ddots & \vdots \\ |D_{K,\theta_0}|^2 & |D_{K,\theta_{U+1}}|^2 & \cdots & |D_{K,\theta_K}|^2 \end{bmatrix}}_{\mathbf{G}(\omega)} \underbrace{\begin{bmatrix} \phi_S \\ \phi_{N_{\theta_{U+1}}} \\ \vdots \\ \phi_{N_{\theta_K}} \end{bmatrix}}_{\mathbf{\Phi}_{S+N}(\omega,t)}, \tag{7}$$

where $\phi_{Z'_\theta}(\omega,t)$ are the PSDs calculated from $Z'_\theta(\omega,t)$ (i.e. outputs of (6)). Note that since rotor noise is removed beforehand, the rotor noise source $N_{\theta_u}(\omega,t)$ and consequently the beamformer output PSD pointing towards it $\phi_{Y_{\theta_u}}(\omega,t)$, need no longer to be considered.

Again, assuming that the measured IRs of each angle of interest are known, the matrix $\mathbf{G}(\omega)$ can be calculated beforehand. The source PSDs can then be estimated as

$$\mathbf{\Phi}_{S+N}(\omega,t) = \mathbf{G}^{-1}(\omega)\mathbf{\Phi}_{Z'_\theta}(\omega,t). \tag{8}$$

Following (8), $\mathrm{WF}_{\mathrm{int}}$ is designed using $\mathbf{\Phi}_{S+N}(\omega,t)$ is utilised to separate the target and coherent interfering noise sources, and thereby extracting the final output signal $Z(\omega,t)$. The weights of $\mathrm{WF}_{\mathrm{int}}$ is given as

$$H(\omega,t) = \frac{\phi_S(\omega,t)}{\phi_S(\omega,t) + \sum_{n=U+1}^{K} \phi_{N_{\theta_n}}(\omega,t)}. \tag{9}$$

Finally, $Z(\omega,t)$ is obtained as

$$Z(\omega,t) = H(\omega,t) Z'_{\theta_0}(\omega,t). \tag{10}$$

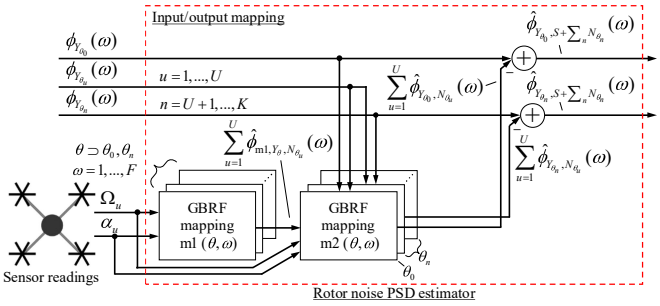


Figure 2: Rotor noise PSD estimation module [9], which contains the GBRF based input/output mapping functions with multi-sensory input information.

2.3 Rotor noise PSD estimator input/output mapping

In the study [9], the multi-sensory, machine learning-based rotor noise PSD estimator uses the gradient boosted random forest mapping function (GBRF). This is a combination of regression tree (RT), gradient boosting (GB), and random forest (RF) [22]. Details of its architecture are well described in common textbooks and in [9]. The GBRFs are prepared for F independent frequency bins and each beamformer output (except those pointing towards the rotors), giving a total of $(1 + K - U) \times F$ GBRF models.

For the input feature used for the GBRFs in [9], two mapping configurations (m1 and m2, see Section 2.3) are used. Common features utilised by both configurations are **rotor speed** $(\Omega_u(t))$ and **rotor acceleration** $(\alpha_u(t))$. It is shown in [9] that rotor noise PSD consists of many tonal harmonics, of which its frequencies follow $\Omega_u(t)$, which means such information can be utilised to infer the rotor noise PSD without concerns from acoustical disturbances.

In addition to $\Omega_u(t)$ and $\alpha_u(t)$, m2 also utilises acoustic signals, specifically **beamformer output PSD** $(\phi_{Y_\theta}(\omega,t))$ and **output of m1** $(\sum_{u=1}^{U} \hat{\phi}_{\mathrm{m1},Y_\theta,N_{\theta_u}}(\omega,t))$. While $\phi_{Y_\theta}(\omega,t)$ contains disturbance from sources other than rotor noise, it is the closest representation of the true rotor noise PSD while the UAV is in operation. Therefore, it could serve as a useful reference to capture the details of rotor noise PSD $\sum_{u=1}^{U} \phi_{Y_\theta,N_{\theta_u}}(\omega,t)$. On the other hand, $\sum_{u=1}^{U} \hat{\phi}_{\mathrm{m1},Y_\theta,N_\theta}(\omega,t)$ provides an undisturbed estimate of the rotor noise PSD was well demonstrated to be an important input feature that supplements $\phi_{Y_\theta}(\omega,t)$ in [9].

More details concerning the sound source localisation method and its performance evaluation through in-flight experiments can be found in [9].

3. Sound source localisation

This section reviews the method from [14].

3.1 Problem setup

For the source localisation problem for UAVs, we slightly modify the problem setup from source enhancement. The UAV utilised in this section is also different from Section 2.. While still assuming a UAV system with a M-sensors, receiving a single target sound source, K interfering *spatially*

coherent noise sources (including U UAV rotors), and ambient *spatially incoherent* noise, the objective of the system here is to accurately locate the target sound source using the M-channel noisy recordings. Therefore, we re-express (1) to explicitly account for a source signal's direction of arrival (DOA) as

$$\begin{aligned}\mathbf{x}(\omega,t) &:= \Big[X_1(\omega,t), \quad \cdots, \quad X_M(\omega,t)\Big]^T \\ &= \mathbf{a}(\omega,\vec{\theta}_S)S(\omega,\vec{\theta}_S,t) \\ &+ \sum_{u=1}^{U}\mathbf{a}(\omega,\vec{\theta}_u)N(\omega,\vec{\theta}_u,t) \\ &+ \sum_{n=U+1}^{K}\mathbf{a}(\omega,\vec{\theta}_n)N(\omega,\vec{\theta}_n,t)+\mathbf{v}(\omega,t), \end{aligned} \tag{11}$$

where $\mathbf{a}(\omega,\vec{\theta}) = [A_1(\omega,\vec{\theta}),\cdots,A_M(\omega,\vec{\theta})]^T$ and $\mathbf{v}(\omega,t)$ are the vector of transfer functions between the source at DOA $\vec{\theta} = [\theta_{\text{el}},\theta_{\text{az}}]^T$ (where el and az indicate the elevation and azimuth directions, respectively) and each microphone m, and the incoherent noise vector observed by the microphone array, respectively. $S(\omega,\vec{\theta}_S,t)$, $N(\omega,\vec{\theta}_u,t)$ and $N(\omega,\vec{\theta}_n,t)$ are the STFT of the target sound source at angle $\vec{\theta}_S$, the noise source coming from the u-th rotor at angle $\vec{\theta}_u$, and the n-th *spatially coherent interfering* noise source at angle $\vec{\theta}_n$, respectively. Like in Section 2.1, we assume $\mathbf{v}(\omega,t)$ is negligible. For the 3D problem, $\vec{\theta}_S$, $\vec{\theta}_u$ and $\vec{\theta}_n$ are expressed in spherical coordinates as

$$\vec{\theta}_S = [\theta_{S,\text{el}},\theta_{S,\text{az}}]^T, \quad \vec{\theta}_u = [\theta_{u,\text{el}},\theta_{u,\text{az}}]^T, \quad \vec{\theta}_n = [\theta_{n,\text{el}},\theta_{n,\text{az}}]^T. \tag{12}$$

The assumptions imposed in Section 2.1 are also applicable to the source localisation problem. In order to identify the directions of the target sound source, knowledge regarding the transfer function $\mathbf{a}(\omega,\vec{\theta})$ is required, which, unfortunately, is generally unavailable. Thus, we assume the UAV operates at some height above ground and is mostly open air. Therefore, the environment is approximately a free field, and that $\mathbf{a}(\omega,\vec{\theta})$ is assumed as the steering vector of a plane wave [6], described as $\mathbf{a}(\omega,\vec{\theta}) = \Big[e^{-j\omega\tau_{\vec{\theta},1}},\cdots,e^{-j\omega\tau_{\vec{\theta},M}}\Big]^T$, where $\tau_{\vec{\theta},m}$ is the TDOA at the m-th microphone with respect to the reference microphone typically placed at the origin of the coordinate. It should be noted that this assumption is merely made for modelling the transfer function between the microphones and the sound source.

The problem in this section is based on the DRone EGonoise and localizatiON (DREGON) database [23], which considers three distinct tasks for the UAV and the target sound source:

Task 1. Hovering UAV - where both the target sound source and UAV are fixed in position throughout the audio recording.

Task 2. Flying (i.e. moving) UAV, broadband sound source - Here, the UAV is assumed to be moving while the target sound source (continuous broadband signal) remains fixed.

Task 3. Flying (i.e. moving) UAV, speech sound source - Here, the UAV is assumed to be moving while the target sound source (speech signal) remains fixed.

We assume that for tasks 2 and 3, the UAV moves relatively smoothly, such that there are no erratic variations in the rotor noise signature. In addition, the DREGON database only contains the target sound source and UAV rotor noise. Thus no additional coherent interfering noise sources exist (i.e. $K = U$).

3.2 Proposed method

Figure 3 shows a block diagram of the localisation method from [14]. The method is based on the method from [24], however with extensions proposed to address the very low SNR unique to the UAV problem setup. We first introduce the baseline method from [24] in Section 3.2.1, followed by the extensions and modifications made to the baseline method.

3.2.1 Multi-source TDOA estimation in reverberant audio using angular spectra

This section outlines the baseline method from [24]. First, the SNR is calculated in the angular TDOA and time-frequency spectrum using pairs of microphones within the array, giving $K_{\text{p}} = {}_M C_2$ unique spectrum, which we refer it as the *SNR response*. Prior to calculating the SNR response, a mapping between a grid of TDOAs τ and a relevant range of $\vec{\theta}$ in the elevation and azimuth plane (i.e. the angular spectra) for each k-th microphone pair is established as follows

$$\tau_k(\theta_{\text{el}},\theta_{\text{az}}) = \frac{p_k \sin\left(\alpha_k\left(\theta_{\text{el}},\theta_{\text{az}}\right)\right)}{c_0}, \tag{13}$$

$$\alpha_k(\theta_{\text{el}},\theta_{\text{az}}) = \cos^{-1}\left(\frac{\mathbf{d}_k\left(\theta_{\text{el}},\theta_{\text{az}}\right)\cdot\Delta\mathbf{p}_k}{p_k}\right), \tag{14}$$

where $\mathbf{d}_k$ is the directional vector associated with angle $\vec{\theta}$ and c_0 is the speed of sound. $\Delta\mathbf{p}_k$ is the separation between the k-th pair of microphones in Cartesian coordinates and p_k is the magnitude of the separating distance. Here, the target sound source is assumed to be located within this angular range of interest.

The baseline method from [24] provides several localisation techniques to calculate the SNR response (hereby denoted as $\psi_k(t,\omega,\tau_k)$ for each k-th microphone pair) for localisation. As such, the study in [14] explored methods such as the delay-and-sum [25] and MVDR beamforming techniques, as well as the generalised cross-correlation - phase transform (GCC-PHAT) [26]. In addition, a non-linear extension of GCC-PHAT proposed in [27], and a modified MVDR approach developed in [24] to improve robustness against diffuse ambient noise, named diffuse noise model, is also utilised in [14].

Following the calculation of $\psi_k(t,\omega,\tau_k)$, the SNR responses are aggregated together across the frequency bins, time frames, and the K_{p} microphone pairs, to give an overall SNR response in terms of $\vec{\theta}$ (i.e. an angular spectrum). Aggregation across the frequency bins and the microphone pairs is carried out via summing, while time frames can

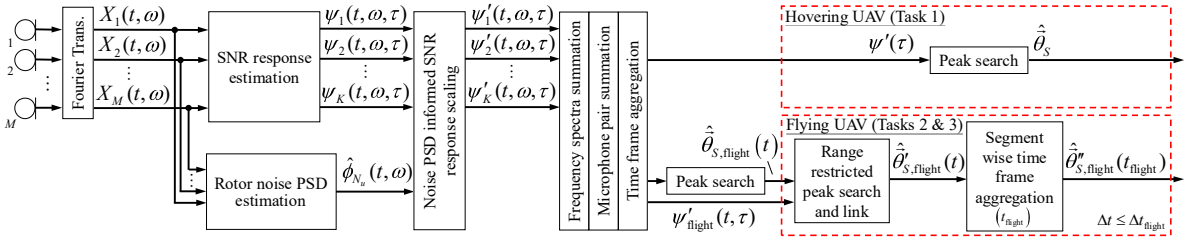


Figure 3: Block diagram of the proposed sound source localisation method in [14].

be summed or taken to the maximum. In [14], time frame aggregation depends on the task. In task 1 (i.e. hovering UAV), all T_{hover} time frames are aggregated to give a single location estimate. This is done as the relative location between the microphone array, and the target sound source remains fixed. Aggregation for task 1 is calculated as

$$\psi'^{\text{sum}}(\tau) = \sum_{t=1}^{T_{\text{hover}}} \sum_{k=1}^{K_{\text{p}}} \sum_{\omega=1}^{F} \psi_k(t, \omega, \tau_k), \tag{15}$$

$$\psi'^{\max}(\tau) = \max_{t}^{T_{\text{hover}}} \sum_{k=1}^{K_{\text{p}}} \sum_{\omega=1}^{F} \psi_k(t, \omega, \tau_k). \tag{16}$$

Subsequently, the overall SNR response for tasks 2 and 3 are calculated as

$$\psi'^{\text{sum}}_{\text{flight}}(t, \tau) = \sum_{t=1}^{T_{\text{flight}}} \sum_{k=1}^{K_{\text{p}}} \sum_{\omega=1}^{F} \psi_k(t, \omega, \tau_k), \tag{17}$$

$$\psi'^{\max}_{\text{flight}}(t, \tau) = \max_{t=1}^{T_{\text{flight}}} \sum_{k=1}^{K_{\text{p}}} \sum_{\omega=1}^{F} \psi_k(t, \omega, \tau_k). \tag{18}$$

The TDOA τ that gives the maximum overall SNR response from $\psi'(\tau)$, denoted $\hat{\tau}_S$ is then calculated as

$$\hat{\tau}_S = \underset{\tau}{\text{argmax}} \left(\psi'(\tau) \right), \tag{19}$$

$$\hat{\tau}_{S,\text{flight}}(t) = \underset{\tau}{\text{argmax}} \left(\psi'_{\text{flight}}(t, \tau) \right). \tag{20}$$

Finally, following (13) and (14) using $\hat{\tau}_S$ and $\hat{\tau}_{S,\text{flight}}(t)$, we obtain the source location in terms of angle for tasks 1 ($\hat{\vec{\theta}}_S$), 2 and 3 ($\hat{\vec{\theta}}_{S,\text{flight}}(t_{\text{flight}})$).

3.2.2 Noise PSD informed SNR response scaling

This section introduces the UAV rotor noise PSD-based weighting envelope to scale and denoise the SNR response $\psi(t, \omega, \tau)$, referred to as *SNR response scaling*.

While it is shown in [9] that adopting a machine learning-based rotor noise PSD estimator enables highly accurate PSD estimation, one of the challenges with the DREGON database is the limited amount of rotor noise data available for training. Therefore, conventional neural networks would not be suitable for the task. On the other hand, denoising autoencoders (DAE) learn a compressed representation of the uncorrupted input rather than a full mapping of the training data. Therefore, it can be used for feature extraction, and denoising [28]. Here, the input of the DAE is the PSDs from the microphone recordings $\phi_m(t, \omega)$ to output an accurate estimate of the rotor noise PSD $\phi_u(\omega)$. This is then used to create an envelope to scale and denoise the SNR response $\psi(t, \omega, \tau)$.

To form the encoder component of the DAE, the input audio PSDs $\phi_m(t, \omega)$ from each microphone are used to map towards the hidden representations z. Subsequently, the rotor noise PSD $\hat{\phi}_u(\omega)$ is reconstructed from z, which forms the decoder component of the DAE.

The size of the input audio PSD data is $T_{\text{DAE}} \times F$, where $T_{\text{DAE}} = 1$ corresponds to the number of PSD frames taken per observation. Details regarding the DAE architecture can be found in [14]. The DAE is optimised with respect to the mean squared error (MSE) between the output PSD $\hat{\phi}_u(t, \omega)$ and the true rotor noise PSD $\phi_u(t, \omega)$. To optimise MSE loss, the Adam optimiser is used [29]. The DAE is trained for each m microphone channel, giving a total of M DAEs for producing the SNR response scaling weighting envelope. To maximise noise removal effectiveness, the estimated PSD with the most prominent amplitude response out of the M microphones for each frequency bin ω is selected and applied to scale the SNR responses for all K_{p} microphone pairs. In addition, the estimated PSD frames are grouped and averaged to match the time frames for the localisation process.

Finally, the rotor noise PSD scaled SNR response is obtained as

$$\psi'_k(t, \omega, \tau) = \frac{\psi_k(t, \omega, \tau)}{\hat{\phi}_u(t, \omega)}. \tag{21}$$

This response then follows the aggregation process (15)-(18) to obtain the overall SNR response, which will be used to calculate $\hat{\vec{\theta}}_S$ using (13) and (14).

3.3 Angular spectral range restricted peak search and link

As discussed in Section 3.2.1, in tasks 2 and 3, time frame aggregation is carried out with smaller groups of frames T_{flight}, which potentially causes a loss in angular spectral resolution. To combat this issue, an angular spectral range restricted peak search and link post-processing algorithm (RPSL) is proposed in [9]. The algorithm is applied towards the localisation output $\hat{\vec{\theta}}_{S,\text{flight}}(t)$ before time frame aggregation is carried out (see Figure 3).

A flowchart describing the algorithm is shown in Figure 4. The algorithm carries out SNR response peak searching in the angular spectrum for several iterations to obtain the correct sound source travel path, which generally follows these main steps:

Step 1. Using localisation output $\hat{\vec{\theta}}_{S,\text{flight}}(t)$ as the reference

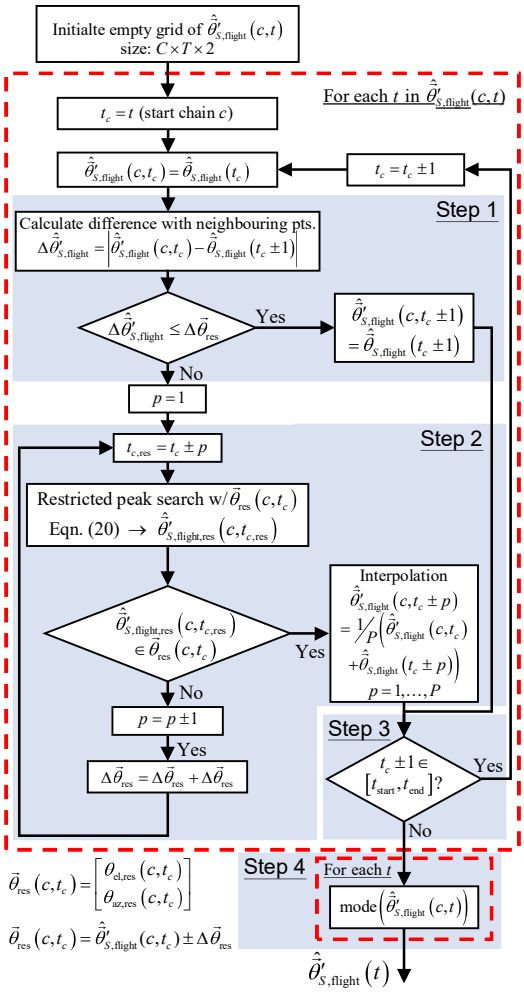


Figure 4: Flowchart of the RPSL algorithm. The **Steps** highlighted follows the descriptions in Section 3.3.

path of locations, check the difference $\Delta\hat{\vec{\theta}}'_{S,\text{flight}}$ between $\hat{\vec{\theta}}_{S,\text{flight}}(t)$ with respect to $\hat{\vec{\theta}}_{S,\text{flight}}(t-1)$ and $\hat{\vec{\theta}}_{S,\text{flight}}(t+1)$ for each time frame t.

Step 2. Perform restricted peak search using (20) with $\psi'_{\text{flight}}(t,\tau)$ (see Section 3.2.2) and $\vec{\theta}_{\text{res}}(c,t_c)$ (see 4) around time frames where $\Delta\hat{\vec{\theta}}'_{S,\text{flight}}$ exceeds threshold $\Delta\vec{\theta}_{\text{res}}$).

Step 3. Repeat Step 1 and Step 2 until no valid locations can be found, or if the start/end of the localisation path is reached (i.e. $t_c \pm 1 \notin [t_{\text{start}}, t_{\text{end}}]$). This forms the c-th "chain" of locations/local path (see Figure 4).

Step 4. Upon obtaining all C chains of local paths, we find the final path $\hat{\vec{\theta}}'_{S,\text{flight}}(t)$ by selecting locations that appear most frequently amongst the C chains at each t-th time frame.

Finally, the T_{flight} time frames in $\hat{\vec{\theta}}_{S,\text{flight}}(t)$ are then aggregated together to obtain $\hat{\vec{\theta}}'_{S,\text{flight}}(t_{\text{flight}})$ (see Figure 3). Note that the threshold $\Delta\hat{\vec{\theta}}_{\text{res}}$ is heuristically tuned based on whether the estimated path of locations appears to be sensible overall (i.e. no aggressive jumps or unnatural changes in direction).

The RPSL post-processing algorithm is carried out in 2 second frame batches of $\hat{\vec{\theta}}_{S,\text{flight}}(t)$, except for the last batch, which would depend on the number of frames remaining. If the restricted peak search fails to obtain a valid location in a particular local path/time frame, the algorithm will skip the time frame and proceed to the next. Following this, the skipped locations are later obtained via interpolation between two valid time frames.

More details concerning the sound source localisation method, and its performance evaluation through experiments using the DREGON database can be found in [14].

4. Conclusions

This article has overviewed sound source enhancement and localisation techniques designed specifically for audio recording systems for UAVs using microphone arrays. Both application results in highly distinct algorithms, where source enhancement is based on beamforming with rotor noise informed postfiltering, and sound source localisation uses an extended multi-source TDOA estimation method. However, both applications share a common trait of having a dedicated means to reduce the influence of rotor noise on their respective received audio.

Regardless, UAV-specific signal processing remains a challenging task and has many aspects open to future studies. Methods to further improve rotor noise reduction and, more importantly, maintain or improve the output audio quality require much research in source enhancement. For source localisation, tracking moving sources more effectively also requires much work. For example, collecting more rotor noise data to better train the rotor noise PSD estimation DAE would drastically improve its performance. Utilising multiple UAVs to triangulate the localisation results (such as that proposed in [13]) would be another viable approach.

References

[1] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*, Digital Signal Processing. Springer, 2001.

[2] K. Furukawa, K. Okutani, K. Nagira, T. Otsuka, K. Itoyama, K. Nakadai, and H. G. Okuno, "Noise correlation matrix estimation for improving sound source localization by multirotor UAV," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 3943–3948.

[3] K. Nakadai, M. Kumon, H. G. Okuno, K. Hoshiba, M. Wakabayashi, K. Washizaki, T. Ishiki, D. Gabriel, Y. Bando, T. Morito, R. Kojima, and O. Sugiyama, "Development of microphone-array-embedded UAV for search and rescue task," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2017, pp. 5985–5990.

[4] K. Yamada, M. Kumon, and T. Furukawa, "Belief-driven control policy of a drone with microphones for multiple

sound source search," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019, pp. 5326–5332.

[5] T. Morito, O. Sugiyama, R. Kojima, and K. Nakadai, "Partially shared deep neural network in sound source separation and identification using a UAV-embedded microphone array," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 1299–1304.

[6] Y. Hioka, M. Kingan, G. Schmid, and K. A. Stol, "Speech enhancement using a microphone array mounted on an unmanned aerial vehicle," in *2016 IEEE International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sept. 2016, pp. 1–5.

[7] L. Wang and A. Cavallaro, "Acoustic sensing from a multi-rotor drone," *IEEE Sensors Journal*, vol. 18, no. 11, pp. 4570–4582, June 2018.

[8] Z-W. Tan, A. H-T. Nguyen, and A. W-H. Khong, "An efficient dilated convolutional neural network for UAV noise reduction at low input SNR," in *2019 Proceedings of Asia-Pacific Signal and Information Processing Association (APSIPA)*, Nov. 2019, pp. 1885–1892.

[9] B. Yen, Y. Hioka, G. Schmid, and B. Mace, "Multi-sensory sound source enhancement for unmanned aerial vehicle recordings," *Applied Acoustics*, vol. 189, pp. 108590, 2022.

[10] Y. Song, S. Kindt, and N. Madhu, "Drone ego-noise cancellation for improved speech capture using deep convolutional autoencoder assisted multistage beamforming," in *2022 25th International Conference on Information Fusion (FUSION)*. IEEE, 2022, pp. 1–8.

[11] Y. Hioka, M. Kingan, G. Schmid, R. McKay, and K. A. Stol, "Design of an unmanned aerial vehicle mounted system for quiet audio recording," *Applied Acoustics*, vol. 155, pp. 423 – 427, 2019.

[12] B. Yen, Y. Hioka, and B. Mace, "Source enhancement for unmanned aerial vehicle recording using multi-sensory information," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 850–857.

[13] T. Yamada, K. Itoyama, K. Nishida, and K. Nakadai, "Assessment of sound source tracking using multiple drones equipped with multiple microphone arrays," *International journal of environmental research and public health*, vol. 18, no. 17, pp. 9039, 2021.

[14] B. Yen and Y. Hioka, "Noise power spectral density scaled snr response estimation with restricted range search for sound source localisation using unmanned aerial vehicles," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2020, no. 1, pp. 1–26, 2020.

[15] W. Manamperi, T. D. Abhayapala, J. Zhang, and P. N. Samarasinghe, "Drone audition: Sound source localization using on-board microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 508–519, 2022.

[16] A. Deleforge, D. Di Carlo, M. Strauss, R. Serizel, and L. Marcenaro, "Audio-based search and rescue with a drone: highlights from the ieee signal processing cup 2019 student competition [sp competitions]," *IEEE Signal Processing Magazine*, vol. 36, no. 5, pp. 138–144, 2019.

[17] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug 1969.

[18] Y. Hioka and K. Niwa, "Estimating power spectral density for spatial audio signal separation: An effective approach for practical applications," *Acoustical Science and Technology*, vol. 38, no. 4, pp. 175–184, 2017.

[19] B. Yen, Y. Hioka, and B. Mace, "Estimating power spectral density of unmanned aerial vehicle rotor noise using multisensory information," in *2018 26th European Signal Processing Conference (EUSIPCO)*, Sep. 2018, pp. 2434–2438.

[20] B. Yen, Y. Hioka, and B. Mace, "Improving power spectral density estimation of unmanned aerial vehicle rotor noise by learning from non-acoustic information," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sep. 2018, pp. 545–549.

[21] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, June 1967.

[22] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1, Springer series in statistics New York, 2001.

[23] M. Strauss, P. Mordel, V. Miguet, and A. Deleforge, "Dregon: Dataset and methods for uav-embedded sound source localization," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.

[24] C. Blandin, A. Ozerov, and E. Vincent, "Multi-source TDOA estimation in reverberant audio using angular spectra and clustering," *Signal Processing*, vol. 92, no. 8, pp. 1950 – 1960, Aug. 2012, Latent Variable Analysis and Signal Separation.

[25] I. McCowan, "Microphone arrays: A tutorial," *Queensland University, Australia*, pp. 1–38, 2001.

[26] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, Aug. 1976.

[27] B. Loesch and B Yang, "Blind source separation based on time-frequency sparseness in the presence of spatial aliasing," *9th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 1–8, 2010.

[28] P. Vincent, H. Larochelle, Y. Bengio, and P-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, Dec. 2014.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-09

深層ブラインド音源分離と転移学習に基づく遠隔音声認識の評価

合澤　隆拓[1,2*]　坂東 宜昭[2]　糸山 克寿[1]　西田 健次[1]　中臺 一博[1]

[1] 東京工業大学　[2] 産業技術総合研究所

概要: 本稿では，深層ブラインド音源分離と転移学習に基づく遠隔音声認識について述べる．複数話者による同時発話の遠隔音声認識には，混合音から各話者の音声を抽出する音源分離が不可欠である．複数話者の遠隔音声認識では，発話区間情報と深層生成モデルを用いた弱教師あり深層ブラインド音源分離が高い性能を発揮することが知られている．しかしこの手法は，混合音を時間周波数クラスタリングした結果を補助特徴量として用いるため，特徴量の計算に時間を要する課題があった．本研究では，より簡便な特徴量で頑健に動作する枠組みの確立を目指す．具体的には，学習データに対するクラスタリング結果を疑似教師として深層生成モデルを事前学習し，学習済みモデルを初期値として深層ブラインド音源分離に転移する．ディナーパーティの遠隔音声認識 (CHiME-6 Challenge) の単語誤り率を評価することで，提案法の有効性を検証した．

1　はじめに

音源分離は，観測信号から個々の音源を抽出する技術で，遠隔音声認識のフロントエンドとして不可欠である [1]．スマートスピーカに代表されるように，単一話者の遠隔音声認識は高い性能を達成しているが，複数話者による会話の遠隔音声認識は，オーバーラップや話者数がフレームごとに変動するといった多くの課題がある [2]．未知環境でも頑健に動作する遠隔音声認識のフロントエンドには，表現力が高く未知環境でも頑健な手法が求められる．

音源やマイクアレイに対する事前の情報を殆ど用いないブラインド音源分離 (BSS) は，遠隔音声認識のフロントエンドとして広く活用されている [1,3,4]．例えば，混合複素角度中心ガウスモデル (cACGMM) [3,5] は，少ない計算量で小さな音源の移動や残響にも耐える手法として広く研究されている．しかし，本手法は線形の生成モデルに基づくため，表現力に限界があった．そこで非線形生成モデルに基づく深層フルランク空間相関分析 (Neural FCA) [6] が提案されている．このモデルは，VAE [7] の枠組みに基づき，混合音から各音源の潜在変数を推定する推論モデルと潜在変数から分離音を再構成する音源生成モデルからなる．Neural FCA は，数値混合音を用いた音声分離 [6] や，遠隔音声認識のフロントエンド [8] として，cACGMM を含む従来の BSS を上回る性能が報告されている．

Neural FCA を遠隔音声認識のフロントエンドに適用した従来研究 [8] では，音源数が既知であると仮定する従来の Neural FCA に対して，各音源の発話区間情報を生成モデルに導入し，音源数が変動する日常会話の認識を行った．本手法は発話区間情報を用いるため弱教師あり Neural FCA と呼ばれ，cACGMM の分離結果を補助特徴量として推論モデルに入力する．本枠組みは，推論時に cACGMM と Neural FCA の 2 つの BSS を実行する必要があるため，比較的大きな計算時間を要すが，cACGMM の分離結果を入力しない場合は大幅に性能が劣化する．これは，混合音のみの特徴量から教師なしで音源分離を学習する問題が難しく，性質の悪い局所解に陥ってしまうためと考えられる．

統計的モデルを用いた教師なし音源分離により推定された信号を疑似教師データとして教師あり学習する手法が提案されている．戸上ら [9] は，混合音から局所ガウスモデル (LGM) に基づく音源分離手法で分離された信号を疑似教師とし，疑似教師信号と分離信号の差分を Kullback-Leibler ダイバージェンスで最小化する学習を提案した．疑似教師あり学習したモデルは，学習データ全体から普遍的な知識を獲得するため，疑似教師そのものよりも高い分離性能を達成している．

本研究では，従来の BSS の分離結果を用いて疑似教師あり学習させた分離モデルから，弱教師あり Neural FCA を転移学習することで，より高い認識性能の実現を目指す．具体的には，混合音に従来の線形 BSS 法である cACGMM [3] を適用し，その分離結果を疑似教師として音源生成モデルとその推論モデルを学習する．学習されたモデルを初期値として弱教師あり Neural FCA を学習することで，混合音のみの特徴量では破綻する課題を解決する．cACGMM の分離結果を従来手法 [8] では推論モデルの入力に補助情報として使っていたが，本研究では疑似教師として用いるため，推論時は cACGMM の計算を削減できる．提案法は，ホームパーティでの会話を収録した CHiME-6 データセット [2] を用いて評価した．

*連絡先：東京工業大学
〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: aizawa@ra.sc.e.titech.ac.jp

2 深層 BSS に基づく音声分離

本研究の基盤となる弱教師あり Neural FCA [8] について説明する．

2.1 問題設定

本手法では，ホームパーティのような N 人が会話している状況で，以下の問題設定により音源分離を行う．

入力: M チャネル混合音 $\mathbf{x}_{ft} \in \mathbb{C}^M$ と話者 $n = 1, \ldots, N$ の時間フレーム t での発話有無 $u_{nt} \in \{0, 1\}$
出力: 話者 n の分離音 $\hat{s}_{nft} \in \mathbb{C}$

ここで，$f = 1, \ldots, F$ および $t = 1, \ldots, T$ はそれぞれ，周波数および時間インデックスを表す．

2.2 生成モデル

弱教師あり Neural FCA では，従来の深層 BSS (Neural FCA) に発話区間変数を導入した生成モデルを定義する．観測混合音 $\mathbf{x}_{ft}$ を以下のように N_{spk} 個の音源信号と N_{noi} 個の雑音信号の和 $s_{nft} \in \mathbb{C}$ $(n = N_{\mathrm{spk}} + 1, ..., N_{\mathrm{spk}} + N_{\mathrm{noi}})$ で表す．

$$\mathbf{x}_{ft} = \sum_{n \in \mathfrak{N}_t} \mathbf{a}_{nf} s_{nft} \tag{1}$$

ただし，$\mathfrak{N}_t = \{n | u_{nt} = 1\} \cup \{N_{\mathrm{spk}} + 1, ..., N_{\mathrm{spk}} + N_{\mathrm{noi}}\}$ は時間 t に存在する音源の集合，$\mathbf{a}_{nf} \in \mathbb{C}^M$ は音源 n のステアリングベクトルである．音源信号のパワースペクトル密度 (PSD) は，ピッチや包絡といった音源の特徴を表す低次元の潜在ベクトル $\mathbf{z}_{nt} \in \mathbb{R}^D$ を用いて以下のような零平均複素ガウス分布で表現する．

$$s_{nft} \sim \mathcal{N}_{\mathbb{C}}\left(0, g_{\theta,nf}(\mathbf{z}_{nt})\right), \tag{2}$$

$$z_{ntd} \sim \mathcal{N}\left(0, 1\right) \tag{3}$$

ここで，$g_{\theta,nf} : \mathbb{R}^D \to \mathbb{R}_+$ は，$\mathbf{z}_{nt}$ から PSD を出力するパラメータ θ を持つ深層ニューラルネットワーク (DNN) である．以上より，観測混合音 $\mathbf{x}_{ft}$ は，以下の多変量複素ガウス分布に従う．

$$\mathbf{x}_{ft} \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \sum_{n \in \mathfrak{N}_t} g_{\theta,nf}(\mathbf{z}_{nt}) \mathbf{H}_{nf}\right) \tag{4}$$

ただし，$\mathbf{H}_{nf} = \mathbf{a}_{nf}\mathbf{a}_{nf}^{\mathsf{H}} \in \mathbb{S}_+^{M \times M}$ は周波数 f における音源 n の空間相関行列 (SCM) である．本稿では，$\mathbf{a}_{nf}$ の比較的小さな変動を許容すため，SCM をフルランクに緩和する．また DNN $g_{\theta,nf}$ は，次節で説明するように，事前に収集した多チャネル混合音と発話区間情報のみから教師なし学習する．

2.3 償却変分推論を用いた弱教師あり学習

弱教師あり Neural FCA では，多チャネル混合音と発話区間から音源モデル $g_{\theta,nf}$ を弱教師あり学習する．なお，モデルは事前学習した後に本学習に転移させるが，詳細は 3 章で述べる．本学習では対数周辺尤度 $\log p_\theta(\mathbf{X}|\mathbf{H}, \mathbf{U})$ を最大化するような音源モデル $g_{\theta,nf}$ を学習する．ただし，$\mathbf{X} \triangleq \{\mathbf{x}_{ft}\}_{n,t=1}^{N,T}$，$\mathbf{H} \triangleq \{\mathbf{H}_{nf}\}_{n,f=1}^{N,F}$，$\mathbf{U} \triangleq \{u_{nt}\}_{n,t=1}^{N,T}$ である．この対数周辺尤度は直接計算困難なので，以下の推論モデル $q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{U})$ を導入した変分償却推論 [7] を行う．

$$q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{U}) = \prod_{n,t,d} \mathcal{N}_{\mathbb{C}}\left(z_{ntd} \mid \mu_{\phi,ntd}(\mathbf{C}), \sigma^2_{\phi,ntd}(\mathbf{C})\right)$$

ただし，$\mathbf{Z} \triangleq \{z_{ntd}\}_{n,t=1}^{N,T}$ は潜在変数の集合を表し，$\mu_{\phi,ntd}(\mathbf{C}) \in \mathbb{R}$ と $\sigma^2_{\phi,ntd}(\mathbf{C}) \in \mathbb{R}_+$ は，特徴量 $\mathbf{C}$ を入力とするパラメータ ϕ を持つ DNN の出力である．特徴量 $\mathbf{C}$ は $\mathbf{X}$ と $\mathbf{U}$ から計算されるが，混合音と cACGMM の分離マスクを入力した場合に，最も良い WER になることが報告されている [8]．

変分償却推論では，学習データに対する以下の変分下限 $\mathcal{L}$ を最大化するように，DNN のパラメータ θ と ϕ，SCM $\mathbf{H}_{nf}$ を同時に最適化する．

$$\mathcal{L} = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{X}|\mathbf{Z}, \mathbf{H}, \mathbf{U})] - \mathcal{D}_{\mathrm{KL}}[q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{U})|p(\mathbf{Z})] \tag{5}$$

第一項は，対数尤度の期待値であり，変分自己符号化器 [7] と同様に以下のように近似される．

$$\mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{X}|\mathbf{Z}, \mathbf{H}, \mathbf{U})] \approx -\sum_{f,t} \log |\mathbf{Y}_{:ft}| - \sum_{f,t} \mathbf{x}_{ft}^{\mathsf{H}} \mathbf{Y}_{:ft}^{-1} \mathbf{x}_{ft} \tag{6}$$

ただし，$\mathbf{Y}_{:ft} = \sum_{n=1}^{N} \mathbf{Y}_{nft} \in \mathbb{S}_+^M$ は，各音源ごとの $\mathbf{Y}_{nft} = g_{\theta,nf}(\mathbf{z}_{nt}^*)\mathbf{H}_{nf} \in \mathbb{S}_+$ の和である． (5) 式の第二項は，潜在変数 z_{ntd} が事前分布から離れないように促す．この式の最大化により，パラメータ θ と SCM $\mathbf{H}_{nf}$ は $\log p_\theta(\mathbf{X}|\mathbf{H}, \mathbf{U})$ を最大化するように，パラメータ ϕ は，$\mathcal{D}_{\mathrm{KL}}[q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{U})|p_\theta(\mathbf{Z}|\mathbf{X}, \mathbf{H}, \mathbf{U})]$ を最小化するように学習される．各 DNN のパラメータは誤差逆伝播法により最適化する．SCM $\mathbf{H}_{nf}$ は，以下の更新則 [10] を繰り返して最適化する．

$$\mathbf{H}_{nf} \leftarrow \mathbf{B}_{nf}^{-\frac{1}{2}} \left(\mathbf{B}_{nf}^{\frac{1}{2}} \mathbf{A}_{nf} \mathbf{B}_{nf}^{\frac{1}{2}}\right)^{\frac{1}{2}} \mathbf{B}_{nf}^{-\frac{1}{2}} \tag{7}$$

$$\mathbf{A}_{nf} \triangleq \mathbf{H}_{nf} \left(\sum_t g_{\theta,nf}(\mathbf{z}_{nt}^*) \mathbf{Y}_{:ft}^{-1} \mathbf{x}_{ft} \mathbf{x}_{ft}^{\mathsf{H}} \mathbf{Y}_{:ft}^{-1}\right) \mathbf{H}_{nf}$$

$$\mathbf{B}_{nf} \triangleq \sum_t g_{\theta,nf}(\mathbf{z}_{nt}^*) \mathbf{Y}_{:ft}^{-1}$$

ただし，$\mathbf{z}_{nt}^* \sim q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{U})$ は潜在ベクトルのサンプルである．本研究では，θ と ϕ を 1 回更新するごとに $\mathbf{H}$ を 5 回更新する．

3　転移学習を用いた深層BSSの拡張

提案法では，従来の線形 BSS による分離結果を疑似教師として学習されたモデルを，2 章で述べた弱教師あり深層 BSS 音源分離 [8] に転移する．

3.1　疑似教師あり学習

2 章で導入した推論モデル q_ϕ と音源生成モデル $g_{\theta,nf}$ を事前学習する．具体的に，は混合音 $\mathbf{X}$ と発話区間変数 $\mathbf{U}$ を入力とする cACGMM の一種である補助付き音源分離法 (GSS) [3] による混合音の分離結果 $s_{nft} \in \mathbb{C}$ を疑似教師として，ネットワークが s_{nft} を模倣するように学習する．GSS は，発話区間変数でマスクされた cACGMM の対数周辺尤度を EM アルゴリズムにより最大化することで，時間周波数ビンをクラスタリングする手法である．

本研究の疑似教師あり学習では，以下の変分下限 $\mathcal{L}_s$ を最大化する．

$$\mathcal{L}_s = \mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{S}|\mathbf{Z})] - \mathcal{D}_{\mathrm{KL}}[q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{U})|p(\mathbf{Z})] \quad (8)$$

ここで，$\mathbf{S} \triangleq \{s_{nft}\}_{n,f,t=1}^{N,F,T}$ は分離音の集合である．本変分下限の第一項の最大化は，再構成音が疑似教師音 $\mathbf{S}$ に近くなることを意味し，第二項は潜在変数の正規化項である．パラメータ θ は $\log p_\theta(\mathbf{S})$ を最大化するように，パラメータ ϕ は，$\mathcal{D}_{\mathrm{KL}}[q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{U})|p_\theta(\mathbf{Z}|\mathbf{S})]$ を最小化するように学習される．第一項については，疑似教師音 $\mathbf{S}$ が (3) 式のような零平均複素ガウス分布に従うと仮定し，以下のように近似される．

$$\mathbb{E}_{q_\phi}[\log p_\theta(\mathbf{S}|\mathbf{Z})] \approx -\sum_{n,f,t} \log y_{nft} - \sum_{n,f,t} \frac{|s_{nft}|^2}{y_{nft}} \quad (9)$$

ただし，$y_{nft} = g_{\theta,nf}(\mathbf{z}_{nt}^*)$ はネットワークが出力する PSD である．式 (9) は再構成音と疑似教師音の板倉斎藤距離 [11] と等価である．GSS の出力はフルランク型最小分散無歪 (MVDR) ビームフォーマ [12] を用いて得るため，分離音のスケールは不定である．そこで，最尤推定値となる以下のスケールを推定 PSD y_{nft} に乗じて学習した．

$$a_{nf} = \frac{1}{T}\sum_t \frac{|s_{nft}|^2}{y_{nft}} \quad (10)$$

3.2　転移学習

学習済みの推論モデル q_ϕ と 音源生成モデル $g_{\theta,nf}$ は 2.3 章の弱教師あり学習に転移させる．具体的には，学習済みモデルの重みを初期値として重みを更新する．エンコーダに入力する特徴量 $\mathbf{C}$ は，混合音と基準マイクと各マイクの位相差であり，分離結果で疑似教師あり学習した代わりに cACGMM 分離マスクは入力しない．

3.3　推論方法

2 回の学習を経て獲得された音源分離 DNN を使い，未知の混合音を分離する．具体的には，混合音 $\mathbf{X}$ と発話区間変数 $\mathbf{U}$ から特徴量 $\mathbf{C}$ を計算し，$\log p_\theta(\mathbf{X}|\mathbf{H},\mathbf{U},\mathbf{Z})$ を最大化するように PSD $g_{\theta,nf}(\boldsymbol{\mu}_{\phi,nt}(\mathbf{C}))$ と SCM $\mathbf{H}_{nf}$ を推定する [6]．分離音 $\hat{s}_{nft}$ は，PSD と SCM から，MVDR ビームフォーマを用いて得る．

4　評価実験

CHiME-6 データセットで提供されている実収録音を用いて提案法を評価した．

4.1　実験設定

CHiME-6 データセットは，複数の家庭で行われたディナーパーティでの音声を記録したもので，各パーティには 4 人の参加者がいる．kitchen，dining，living からなる室内で 5 または 6 台の 4 チャネルマイクアレイ (Microsoft Kinect v2) で収録され，一つのエリアに少なくとも 2 つのマイクアレイが設置されている．train，dev 及び eval セットに分割され，収録時間はそれぞれ 40 時間 33 分，4 時間 27 分及び 5 時間 12 分である．各録音は 16 kHz で収録されている．

提案法のネットワークアーキテクチャは [8] と同一の設計とした．スペクトログラムは短時間フーリエ変換によって求め，窓長 1024，ホップ長 256 とした．音源の数 N_{spk} は 4，潜在変数 D_{spk} は 50 次元とし，雑音源の数 N_{noi} は 2，潜在変数 D_{noi} が 20 次元とした．式 (5) と (8) の KL 項の重みを周期的に変動させる KL アニーリング [6] を行った．学習は，200 エポックとした．計算量を減らすため，混合信号の全 24 チャンネルのうち，最もパワーの大きい 12 チャンネルで学習を行った．推論モデルに入力する特徴量 $\mathbf{C}$ は，提案法では，基準マイクロホンと他のマイクロホン間のチャネル間位相差と，混合音の対数パワースペクトログラムである．

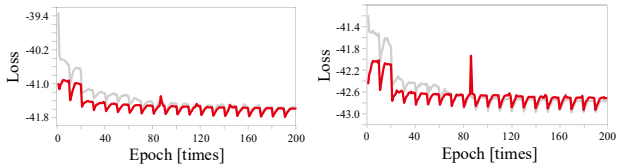
提案法 (WS Neural FCA + 転移学習) は，CHiME-6 Challenge のベースライン音声認識器 [2] を用いて単語誤り率 (WER) で評価した．ベースラインとして，GSS および，GSS を特徴量として用いた場合 (cACGMM → WS Neural FCA) ，事前学習を行わなかった場合 (WS Neural FCA) を評価した．

4.2　実験結果

表 1 に音声認識性能を WER で示す．転移学習した場合，しない場合と比較して 200 エポック目において dev セットで 1.5 pt，eval セットで 0.6 pt 性能が向上

表 1: CHiME-6 データセットの dev set および eval set における WER.

手法	Epoch	Dev set				Eval set			
		Avg.	Dining	Kitchen	Living	Avg.	Dining	Kitchen	Living
GSS (公式実装)	–	51.8	53.8	53.9	48.6	51.3	44.7	61.2	50.3
GSS ($M = 16$) [8]	–	49.8	51.6	52.3	46.4	51.1	45.0	60.8	49.7
GSS → WS Neural FCA [8]	200	48.6	51.2	50.8	45.1	49.0	43.2	56.7	48.9
WS Neural FCA	50	54.8	56.0	58.1	51.0	52.7	45.7	60.2	54.1
WS Neural FCA	100	55.7	56.7	59.6	51.6	52.9	45.7	60.6	54.5
WS Neural FCA	200	55.3	56.2	58.9	51.3	52.9	46.0	60.4	54.1
WS Neural FCA + 転移学習	50	54.1	55.4	57.9	49.9	52.4	45.3	60.3	53.5
WS Neural FCA + 転移学習	100	54.3	55.1	58.1	50.2	52.4	45.2	60.4	53.6
WS Neural FCA + 転移学習	200	53.8	54.8	57.5	49.7	52.3	45.0	60.0	53.8



(a) 学習損失　(b) 検証損失

図 1: 転移学習した場合 (赤) としない場合 (灰) の学習および検証データに対する損失関数.

した．また，図 1 に示す通り，転移学習しない場合は損失関数の収束に 100 エポック程度要しているのに対して，した場合は 50 エポック程度で収束している．

一方，GSS や GSS → WS Neural FCA の結果と比較すると，提案法には改善の余地がある．本稿では，疑似教師の非歪性を重視し，MVDR ビームフォーマの結果を用いたが，ビームフォーマでは妨害音の抑制に限界がある．音源モデルの事前学習においては，MVDR ではなく，時間周波数マスキングの結果を用いた方が性能改善に寄与する可能性がある．また，収束速度は早くなっているが，収束先は転移学習していない場合と大きな差がなく (図 1)，本稿で用いた入力特徴量では効果的な学習が困難な可能性がある．今後は，容易に計算できる学習しやすい特徴量の設計を進める．

5　おわりに

本稿では，従来の BSS の分離結果を用いて疑似教師あり学習したモデルを，弱教師あり Neural FCA に転移学習する枠組みについて述べた．転移学習した場合，しなかった場合と比較して WER がわずかに改善し，損失関数の収束が早いことが示された．WER の大きな改善につながらなかったため，事前学習に用いる疑似教師データや推論モデルへの入力特徴量の改善を進める．

謝辞 本研究の一部は，NEDO および JST ACT-X 数理・情報のフロンティア JPMJAX200N の支援を受けた．

参考文献

[1] K. Shimada *et al.*, “Unsupervised speech enhancement based on multichannel NMF-informed beamforming for noise-robust automatic speech recognition,” *IEEE/ACM TASLP*, vol. 27, no. 5, pp. 960–971, 2019.

[2] S. Watanabe *et al.*, “CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings,” in *CHiME 2020 Workshop*, 2020, pp. 1–7.

[3] C. Boeddeker *et al.*, “Front-end processing for the CHiME-5 dinner party scenario,” in *CHiME5 Workshop*, 2018, pp. 1–6.

[4] K. Sekiguchi *et al.*, “Fast multichannel nonnegative matrix factorization with directivity-aware jointly-diagonalizable spatial covariance matrices for blind source separation,” *IEEE/ACM TASLP*, vol. 28, pp. 2610–2625, 2020.

[5] N. Ito *et al.*, “Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1153–1157.

[6] Y. Bando *et al.*, “Neural full-rank spatial covariance analysis for blind source separation,” *IEEE SPL*, vol. 28, pp. 1670–1674, 2021.

[7] D. P. Kingma *et al.*, “Auto-encoding variational Bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[8] Y. Bando *et al.*, “Weakly-Supervised Neural Full-Rank Spatial Covariance Analysis for a Front-End System of Distant Speech Recognition,” in *Proc. Interspeech 2022*, 2022, pp. 3824–3828.

[9] M. Togami *et al.*, “Unsupervised training for deep speech source separation with kullback-leibler divergence based probabilistic loss function,” in *IEEE ICASSP*, 2020, pp. 56–60.

[10] K. Yoshii, “Correlated tensor factorization for audio source separation,” in *IEEe ICASSP 2018*, 2018, pp. 731–735.

[11] F. Itakura, “Analysis synthesis telephony based on the maximum likelihood method,” *ICA*, 1968.

[12] M. Souden *et al.*, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE TASLP*, vol. 18, no. 2, pp. 260–276, 2010.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-10

Blockwiseストリーミング音声認識と発話区間検出の統合

Integration of Blockwise Streaming Automatic Speech Recognition with Voice Activity Detection

周藤 唯[1*]　Muhammad Shakeel[1]　中臺 一博[2]　史 嘉彤[3]　渡部 晋二[3]
Yui Sudo[1]　Muhammad Shakeel[1]　Kazuhiro Nakadai[1]　Jiatong Shi[3]　Shinji Watanabe[1]

[1] (株) ホンダ・リサーチ・インスティチュート・ジャパン
[1] Honda Research Institute Japan Co., Ltd.
[2] 東京工業大学　[3] カーネギーメロン大学
[2] Tokyo Institute of Technology　[3] Carnegie Mellon University

Abstract: 本稿では，ストリーミング音声を入力とする音声認識アプリケーションに対応するため，Blockwise ストリーミング音声認識と発話区間検出の統合を扱う．近年，エンドツーエンド音声認識は実用的なシステムとして有望視されているが，ストリーミング音声入力に対応するためには以下の課題がある．1) 多くのエンドツーエンド音声認識モデルは音声入力があらかじめ短い発話に区切られていることを前提としているため，前段に発話区間検出モジュールが必要であり，システム全体のパラメータ数が増加する．2) 発話区間検出モジュールによる音声切り出しが適切でない場合，音声認識の性能が劣化する．3) 非発話区間が誤って発話として検出されると，性能劣化に加えて余分なデコードを行うための計算コストが増加する．そこで，本研究では，システム全体のパラメータを削減するため，Blockwise ストリーミング音声認識に発話区間検出ブランチを統合したモデルを提案する．また，ブロックごとに推定される発話区間検出結果を適切にデコード時に利用するため，Re-blocking 処理を提案する．提案手法は，既存の統合手法に対して，パラメータ数の増加を 1%未満に抑えながら，発話区間検出エラー率を 70.1%減少させた．さらに，同等のリアルタイムファクタ (RTF) を維持しつつ，文字誤り率 (CER) を 14.5%改善することができた．

1　はじめに

エンドツーエンド音声認識が実用的なシステムとして盛んに研究されている [1, 2, 3, 4, 5]．エンドツーエンド音声認識システムでは，従来の音声認識システムにおける音響モデルや言語モデル，発音辞書などのコンポーネントが不要になるため，システム構成を単純化することができる．システム構成が単純であることは，運用や保守を簡易化することが可能なため，実用性においては重要な特徴である．一方，ユーザにとっての実用性向上のための重要な要素は遅延時間である．ストリーミング音声を入力するアプリケーションでは，発話の完了を待たずに逐次的に音声認識結果が出力されることが要求されるため，ストリーミング処理が可能なエンドツーエンド音声認識モデルの研究が進められてきた．

最も単純なストリーミング音声認識の手法は，単方向 LSTM (Long Short Term Memory) を用いることである [6]．単方向 LSTM は未来の情報を必要としないため，発話の完了を待たずに認識処理を開始することができる．また，Transformer や Conformer といったオフラインベースの手法に対し，因果的な制約を設けることでストリーミング処理を可能にした Causal Transformer/Conformer も提案されている [7]．もう一つの方向性は，ブロック単位の処理を行うことで，Transformer や Conformer における self-attention の窓長を制限することである．このアプローチは，隠れマルコフモデルに基づくシステムや [8, 9]，RNN (Recurrent Neural Network) トランスデューサ [10, 11]，attention [12]，CTC (Connectionist Temporal Classification)/attention [13, 14, 15, 16] などで広く実現されている．ただし，これらのストリーミング音声認識モデルでは，入力音声があらかじめ発話ごとに分割されていることが前提となっているため，通常，音声認識の前段に発話区間検出モジュールが必要である．

発話区間検出に関しては，エネルギベースの手法 [17]，隠れマルコフモデル [18]，混合ガウスモデル [19] などの統計モデルを用いた手法が提案されている．また，近

*連絡先：(株) ホンダ・リサーチ・インスティチュート・ジャパン
〒 351-0188 埼玉県和光市本町 8-1
E-mail: yui.sudo@jp.honda-ri.com

年では，多層パーセプトロン [20]，LSTM [21, 22]，畳み込みニューラルネットワーク [23]，Transformer [24] を用いた深層学習ベースの方法も提案されている．発話区間検出は比較的計算量が少ないが，モジュールを追加することでシステム全体の複雑さが増してしまう．そこで，発話区間検出をエンドツーエンド音声認識に統合する試みもなされている [25, 26]．CTC ベースの発話区間検出 [27, 28] では，CTC が出力する blank ラベルを非音声セグメントと見なして発話区間の検出を行う．しかし，雑音などの非音声セグメントを明示的に blank ラベルに対応付けて損失関数を定義していないため，雑音を含む非音声に対して性能が劣化することが考えられる．

そこで，本研究では，発話区間検出モジュールの追加によるシステム全体のパラメータ増加を防ぐため，CTC/attention ベースの Blockwise ストリーミング音声認識 [16] に発話区間検出ブランチを統合したモデルを提案する．提案手法では，パラメータ数を最小限に抑えるため，音声認識と発話区間検出はエンコーダを共有する．また，ブロックごとに推定される発話区間検出結果を適切にデコード時に利用するための Re-blocking 処理を提案する．なお，本稿は，[29] の提案手法をもとに，既存の統合手法との比較実験を追加した．

2 ストリーミング音声認識

本節では，Transformer を用いた CTC/attention ベースの音声認識 [5] について述べた後，その拡張であり，提案手法に使用する Blockwise ストリーミング音声認識 [15, 16] について説明する．

2.1 Transformer を用いた音声認識

Transformer を用いた CTC/attention ベースの音声認識は，エンコーダ/デコーダ構造を持つ attention ベースの音声認識モデル [3] に，CTC を追加したモデルである [5]．

エンコーダは，畳み込み層，線形射影層，位置符号化層，Transformer ブロックから構成される．畳み込み層は，式 (1) のように長さ T の音響特徴列 $\boldsymbol{X} = [x_1, ..., x_T]$ を $\boldsymbol{u} = [u_1, ..., u_L]$ にダウンサンプリングする $(L < T)$.

$$\boldsymbol{u} = \mathrm{ConvSubsamp}(\boldsymbol{X}), \tag{1}$$

ダウンサンプリングされた特徴ベクトル列 $\boldsymbol{u}$ は，式 (2) のように，Transformer ブロックによって長さ L の隠れ状態ベクトル $\boldsymbol{h} = [h_1, ..., h_L]$ に変換される．

$$\boldsymbol{h} = \mathrm{TrEncoder}(\boldsymbol{u}), \tag{2}$$

Transformer ブロックは残差接続を持ち，multi-head self-attention 層，全結合層，layer 正規化層からなる．

デコーダは，エンコーダの出力する隠れ状態ベクトル $\boldsymbol{h}$ と過去に推定されたテキスト列 $\boldsymbol{y}_{s-1} = (y_0, ..., y_{s-1})$ を用いて，式 (3) に示すように，s 番目のテキスト y_s を再帰的に推定する．

$$y_s = \mathrm{TrDecoder}(\boldsymbol{h}, \boldsymbol{y}_{s-1}), \tag{3}$$

過去のテキスト列である $\boldsymbol{y}_{s-1}$ は，まず埋め込み特徴に変換される．埋め込み特徴と隠れ状態ベクトル $\boldsymbol{h}$ はデコーダに入力され，線形射影層，ソフトマックス関数を用いて y_s の予測確率が推定される．デコーダは self-attention，source-target attention，position-wise 全結合層から構成される．

CTC/attention ベースのモデルでは，上記に加えて全結合層およびソフトマックス関数から構成される CTC を持つ．CTC は，式 (2) によって出力される隠れ状態ベクトルを入力し，blank ラベルを含むテキスト確率を各時刻ごとに推定する．

2.2 Blockwise Transformer エンコーダ

音声がストリーミングで入力されるアプリケーションでは，音声認識モデルは発話の完了を待たずに逐次的に認識処理を行う必要がある．Transformer でこのようなオンライン処理を実現するため，Blockwise Transformer エンコーダ [15] では，式 (2) のエンコーダへの入力を式 (4) に示すようにブロック単位で行う．

$$\boldsymbol{u}_b = (u_{(b-1)L_{\mathrm{hop}}+1}, ..., u_{(b-1)L_{\mathrm{hop}}+L_{\mathrm{block}}}), \tag{4}$$

$\boldsymbol{u}_b$ は b 番目のブロックの特徴ベクトル列，L_{block}，L_{hop} はブロックサイズとホップ長を表す．b 番目の隠れ状態ベクトル $\boldsymbol{h}_b$ は，予め決められたブロックサイズ L_{block}，ホップ長 L_{hop} に基づいて，式 (5) に示すように，長さ L_{block} の隠れ状態ベクトルに変換される．

$$\boldsymbol{h}_b = \mathrm{BlockTrEncoder}(\boldsymbol{u}_b). \tag{5}$$

2.3 Blockwise ビームサーチ

Blockwise ストリーミング音声認識では，デコーダはエンコーダのブロック処理と同時に認識処理を行う必要がある [16]．attention ベース音声認識のオンラインビームサーチは，隠れ状態ブロック $\boldsymbol{h}_{1:b} = [\boldsymbol{h}_1, ..., \boldsymbol{h}_b]$ が与えられた時における，もっとも確率の大きいテキスト列 $\hat{\boldsymbol{y}}$ を探索する問題として，式 (6) のように表される．

$$\hat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \mathcal{V}^*}{\mathrm{argmax}}(\log(p(\boldsymbol{y}|\boldsymbol{h}_{1:b}))), \tag{6}$$

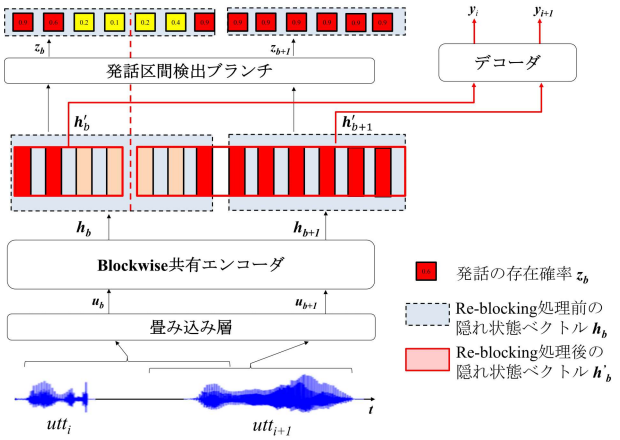


図 1: 提案手法の概要

式 (6) より，テキスト列の推定には，発話の開始から現在のブロックまでのすべての隠れ状態が必要となる．すなわち，発話が長くなるとデコードに要する時間は長くなる．したがって，隠れ状態ベクトル列はデコード前に適切に短い発話に分割し，隠れ状態ベクトルの履歴をリセットする必要がある．

3 提案手法

図 1 に提案手法の概要を示す．提案手法は，エンコーダ/デコーダ構造を持つ既存の CTC/attention ベースの Blockwise ストリーミング音声認識モデル [16] に加えて，発話区間検出ブランチを持つ．発話区間検出ブランチは，既存の音声認識モデルとエンコーダを共有するため，パラメータの増加を抑えることができる．

提案する統合モデルの学習は 2 段階で行う．まず，従来の音声認識モデルを単独で学習し，その後，音声認識モデルのパラメータを固定させて発話区間検出ブランチのみを学習する．推論時には，検出された発話区間情報に応じて適切に隠れ状態列を分割するため，Re-blocking 処理を用いてブロックサイズを調整する．

3.1 発話区間検出ブランチ

発話区間検出ブランチは全結合層とシグモイド関数で構成され，式 (5) の共有エンコーダの出力である隠れ状態ベクトル列 $\boldsymbol{h}_b$ をブロックごとに入力し，フレームごとの発話の存在確率 $\boldsymbol{z}_b = [z_1, ..., z_{L_{\text{block}}}]$ を推定する．時刻 t における発話の存在確率 z_t に対して，閾値 $p = 0.5$ 以上であれば，発話が存在するとみなす．また，発話中に存在する非常に短い非音声セグメントを誤検出しないように，連続する非音声セグメントに対する閾値 V を導入する．連続する非音声セグメントが閾値 V を超えた場合，そのセグメント列は非音声区間とみなされる．第 2.3 節で述べたように，デコード処理に用いる隠れ状態ベクトルが長くなると計算コストが増加してしまうため，非音声区間が検出された場合，現在ブロックまでに得られた隠れ状態ベクトルのデコード処理が完了した後，過去の隠れ状態ベクトルは破棄される．損失関数は，データセットから提供される発話区間情報をもとに目標出力 $\boldsymbol{z}$ を定義し，推定された出力 $\hat{\boldsymbol{z}}$ とのバイナリクロスエントロピーを用いる．

3.2 Re-blocking 処理

第 2.2 節で述べたように，Blockwise ストリーミング音声認識では隠れ状態ベクトル列はあらかじめ決められたサイズのブロック単位で処理される．しかし，非音声セグメントがブロックの途中に存在する場合，隠れ状態列を分割することができない．図 1 に，第 3.1 節で導入した連続する非音声セグメントの閾値が $V = 4$ の場合の具体例を示す．発話区間検出ブランチは，ブロック単位で L_{block} 個の隠れ状態ベクトル $\boldsymbol{h}_b$ を入力し，L_{block} 個の発話の存在確率 $\boldsymbol{z}_b$ を出力する．図 1 における $\boldsymbol{z}_b$ には，黄色で示す 4 個の連続した非音声セグメントが含まれている．連続する非音声セグメントが $V = 4$ 以上であるため，これは非発話区間とみなされるが，$\boldsymbol{z}_b$ の末尾のフレームには赤で示される次の発話冒頭の音声セグメントが含まれている．そのため，単純に長さ L_{block} のブロック単位で隠れ状態ベクトルを分割し，履歴をリセットしてしまうと次の発話の文脈が途切れてしまう．

そこで，デコード時には，検出された非音声セグメントの中心である C 番目の隠れ状態ベクトルを中心に，式 (7) に示すようにブロックサイズを変更 (Re-blocking 処理) する．

$$\boldsymbol{h}'_b = [h_{b_1}, ..., h_{b_C}]. \tag{7}$$

ここで，h_{b_1}，h_{b_C} は b 番目のブロックの先頭，C 番目の隠れ状態ベクトルを表す ($C < L_{block}$)．b 番目のブロックの中で，未使用の隠れ状態ベクトルは，次の式のように，次のブロックの隠れ状態ベクトル列 $\boldsymbol{h}_{b+1}$ に結合される．

$$\boldsymbol{h}'_{b+1} = \text{concat}([h_{b_{C+1}}, ..., h_{b_E}], \boldsymbol{h}_{b+1}), \tag{8}$$

h_{b_E} は，b 番目のブロックにおける末尾の隠れ状態ベクトルを表す．

4 評価実験

本節では，提案手法の発話区間検出の性能，音声認識性能向上に対する効果，計算時間の評価を行う．

4.1 実験条件

提案した統合モデルの入力は，サンプリング周波数16kHz，窓長512サンプル，ホップ長128サンプル，80次元のメルフィルタバンク特徴量を用い，SpecAugment [30] を適用した．エンコーダは，ストライドがそれぞれ2，3の2層の畳み込み層，512次元の線形射影層，位置符号化層，12層のTransformerブロックとlayer正規化層から構成される．デコーダは，2048個の隠れユニットを持つ6層のTransformerブロックで構成される．attentionの次元は256で，4つのmultihead self-attentionを持つ．第2.2節で述べたブロックサイズ L_{block} とホップ長 L_{hop} はそれぞれ40と16とした．発話区間検出ブランチには1層の全結合層を用いた．第3.1節で導入した連続する非音声セグメントの閾値は，V=10とした．音声認識モデル部のパラメータ数は30.3Mであったのに対し，統合した発話区間検出ブランチのパラメータ数はわずか0.8Kであった．

第1段階の音声認識モデル学習では，CTC，attentionの損失の重みをそれぞれ0.3，0.7として，マルチタスク学習を行った [4]．第1段階では，学習率0.005，ウォームアップステップ25,000で，音声認識モデル部のみをAdamを用いて40エポック学習した．第2段階では，学習率0.00001，ウォームアップステップ数10,000で，Adamを用いて発話区間検出ブランチを30エポック学習した．音声認識ツールキットとしてESPnetををを使用した [31]．評価には，日本語話し言葉コーパス (CSJ) [32] を使用した．CSJコーパスは，20分以上の長時間の録音を含んでいるため，ストリーミング音声認識の評価に適している．

4.2 発話区間検出タスクにおける評価

発話区間検出タスクにおいて，提案手法を外付けの発話区間検出モデルおよびCTCベースの発話区間検出 [27] と比較し，適切に発話区間を検出できることを確認した．

発話区間検出システムは，発話区間検出エラー率 (ER) を用いて評価した．ERは以下の式で求める．

$$ER = \frac{\sum_{t=1}^{T} F(t) + \sum_{t=1}^{T} M(t)}{\sum_{t=1}^{T} N(t)}, \tag{9}$$

$F(t)$ は誤検出された非音声セグメントの数，$M(t)$ は，検出されなかった音声セグメントの数を表す．

表1に発話区間検出に必要なパラメータ数とERを示す．外付けの発話区間検出モデルは4.45Mのパラメータを必要とするのに対し，提案手法は音声認識のエンコーダを共有するため，発話区間検出ブランチに必要なパラメータ数はわずか0.8Kであった．一方，CTCベースの発話区間検出は，音声認識モデルにおけるCTCによるblank出力を利用するため，パラメータの増加はない．しかし，雑音などの非音声セグメントを明示的にblankラベルに対応付けて損失関数を定義していないため，ERは大きかった．それに対し，提案手法では，追加パラメータ数をわずか1%未満に抑えつつ，CTCベースの発話区間検出手法よりもERを削減した．

表 1: 発話区間検出タスクにおける検出誤り率

手法	パラメータ数	eval1	eval2	eval3
外付け	4.45 M	4.9	3.9	5.4
CTC ベース	0	18.3	17.5	21.8
提案手法	0.8 K	5.5	4.6	6.9

4.3 音声認識タスクにおける評価

次に，提案手法を音声認識性能に対して，以下の手法との比較評価を行った．

- 手動発話切り出し: データセットが提供する，手動で切り出された発話時間情報に従って入力音声を分割した．これは，提案手法の上限値として扱う．
- 外付け発話区間検出: 外付けのBlockwiseストリーミングTransformerを用いた発話区間検出に基づいて，自動的に音声入力を分割した．
- 発話区間検出なし: 発話区間検出モジュールを用いず，隠れ状態ベクトル列の長さが L_{th}=300を超えた場合に自動的に音声入力を分割した．
- CTCベース発話区間検出 [27]: 連続するCTCのblank出力を非音声セグメントと見なし，自動的に分割した．この手法はもともとLSTMベースのエンコーダを使用していたが，公平な比較のためBlockwiseストリーミングTransformerに拡張した．
- 提案手法: 提案手法に基づき音声入力を自動分割した．また，3.2節で述べた発話区間検出の結果に基づいて，隠れ状態ブロックをRe-blocking化した．また，まれに長時間の発話があるため，すべての手法において，隠れ状態ベクトル列の長さが L_{th}=300を超えた場合にも自動的に音声を分割した．

認識性能の評価指標には，文字誤り率 (CER) を用いた．また，計算コストを検証するため，GPU (NVIDIA A100-SXM4-40GB) を用いて推論を行った際のリアルタイムファクタ (RTF) を測定した．

表 2: CER と RTF の比較

手法	eval1	eval2	eval3	RTF
手動発話切り出し	5.9	4.2	4.6	N/A
外付け発話区間検出	8.0	5.4	6.6	N/A
発話区間検出なし	10.1	7.4	7.9	0.55
CTC ベース	10.3	7.8	9.4	**0.40**
提案手法	**9.1**	**6.7**	**7.7**	**0.40**

4.3.1 実験結果

表 2 に CER と RTF を示す．まず，発話区間検出を用いない場合と提案手法を比較すると，すべての条件において CER と RTF を同時に削減することができた．発話区間検出なしの場合，発話の区切りとは無関係に一定間隔 (L_{th}=300) で音声が分割されるので，発話のコンテキストが分断されてしまうのに対し，提案手法では，発話区間検出ブランチを用いて適切に発話が分割されるため，CER が改善した．

RTF が改善した要因は，適切なタイミングで隠れ状態ベクトルの履歴をリセットすることで，計算量が削減されたことである．図 2 にデコード時の隠れ状態ベクトルのメモリサイズを示す．発話区間検出を用いない場合，非音声区間であってもデコード処理を続けてしまう．また，発話が非常に短かったとしても，一定時間 (L_{th}=300) の履歴を保持してしまう．それに対し，提案手法では，適切な発話の区切りで隠れ状態ベクトルの履歴をリセットすることができるため，RTF を削減することができた．

CTC ベースの手法と提案手法を比較すると，同等の RTF を維持したまま CER を改善することができた．表 1 に示すように，提案手法ではわずか 0.8K のパラメータしか追加していないため，計算コストへの影響はほとんど見られなかった．したがって，提案手法は，既存の CTC ベースの統合手法に対し，計算コストをほとんど増加させることなく，CER を改善することができたといえる．

しかし，提案手法は，外付けの発話区間検出を用いた場合よりも CER が悪化した．外付けの発話区間検出では，エンコーダも含めたモデル全体を同時に最適化したのに対し，提案手法は事前に学習された音声認識モデルのエンコーダを固定した状態で発話区間検出ブランチのみを最適化したため，性能が劣化したことが考えられる．今後の課題として，提案した統合モデルにおいても，2 段階学習ではなく発話区間検出ブランチと音声認識部を同時に最適化する手法の検討が必要であると考える．

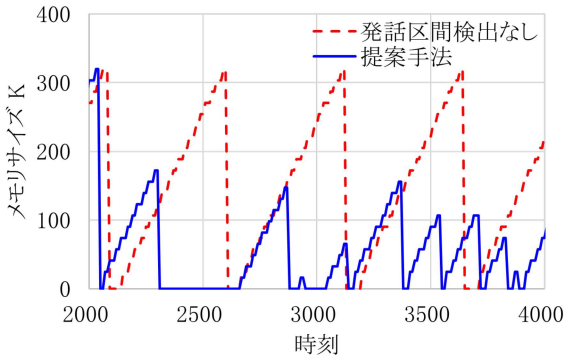


図 2: 隠れ状態ベクトルのメモリサイズ比較

表 3: Re-blocking の効果

手法	eval1	eval2	eval3
発話区間検出なし	10.1	7.4	7.9
Re-blocking なし	10.4	7.7	11.7
Re-blocking あり	**9.1**	**6.7**	**7.7**

4.3.2 Re-blocking の効果

表 3 に Re-blocking 処理の効果を示す．Re-blocking 処理を用いることで CER が改善したが，反対に Re-blocking 処理を用いない場合，むしろ発話区間検出なしの場合よりも CER は悪化した．これは，第 3.2 節で述べたように，Re-blocking 処理を用いずに事前に決められたブロック単位で音声を分割し、履歴をリセットしてしまうと，次の発話の文脈が途切れてしまうことがあるためである．したがって，Blockwise ストリーミング音声認識モデルにおいては，発話の文脈が不適切に分断されることを防止するため，提案した Re-blocking 処理が必要であることがわかった．

5 結論

本研究では，システム全体のパラメータを削減するため，Blockwise ストリーミング音声認識に発話区間検出ブランチを統合したモデル，および，ブロックごとに推定される発話区間検出結果を適切にデコード時に利用するための Re-blocking 処理を提案した．提案手法は，CTC ベースの既存の統合手法に対して，パラメータ数の増加を 1%未満に抑えながら，発話区間検出エラー率を 70.1%減少させた．さらに，同等のリアルタイムファクタ (RTF) を維持しつつ，文字誤り率 (CER) を 7.5%改善することができた．今後は，発話区間検出ブランチと音声認識部を同時に最適化することで，音声認識性能をさらに向上させる予定である．

参考文献

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[2] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML*, 2012.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[4] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[5] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019, pp. 449–456.

[6] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. ASRU*, 2017, pp. 193–199.

[7] B. Li, A. Gulati, J. Yu, T. N. Sainath, C.-C. Chiu, A. Narayanan, S.-Y. Chang, R. Pang, Y. He, J. Qin *et al.*, "A better and faster end-to-end model for streaming asr," in *Proc. ICASSP*, 2021.

[8] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for ASR," in *Proc. ICASSP*, 2018, pp. 5874–5878.

[9] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *Proc. ICASSP*. IEEE, 2021.

[10] L. Lu, C. Liu, J. Li, and Y. Gong, "Exploring transformers for large-scale speech recognition," *Proc. Interspeech*, pp. 5041–5045, 2020.

[11] Y. Shi, V. Nagaraja, C. Wu, J. Mahadeokar, D. Le, R. Prabhavalkar, A. Xiao, C.-F. Yeh, J. Chan, C. Fuegen *et al.*, "Dynamic encoder transducer: A flexible solution for trading off accuracy for latency," *arXiv preprint arXiv:2104.02176*, 2021.

[12] R. Fan, P. Zhou, W. Chen, J. Jia, and G. Liu, "An online attention-based model for speech recognition," *Proc. Interspeech*, pp. 4390–4394, 2019.

[13] N. Moritz, T. Hori, and J. Le, "Streaming automatic speech recognition with the transformer model," in *Proc. ICASSP*, 2020, pp. 6074–6078.

[14] M. Li, C. Zorilă, and R. Doddipatla, "Head-synchronous decoding for transformer-based streaming asr," in *Proc. ICASSP*, 2021.

[15] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer ASR with contextual block processing," in *Proc. ASRU*, 2019, pp. 427–433.

[16] E. Tsunoo, Y. Kashiwagi, and S. Watanabe, "Streaming transformer ASR with blockwise synchronous beam search," in *Proc. SLT*, 2021, pp. 22–29.

[17] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[18] D. Haws, D. Dimitriadis, G. Saon, S. Thomas, and M. Picheny, "On the importance of event detection for ASR," in *Proc. ICASSP*, 2016, pp. 5705–5709.

[19] A. Lee, K. Nakamura, R. Nisimura, H. Saruwatari, and K. Shikano, "Noice robust real world spoken dialogue system using GMM based rejection of unintended inputs," in *Proc. ICSLP*, 2004.

[20] N. Ryant, M. Liberman, and J. Yuan, "Speech activity detection on YouTube using deep neural networks," in *Proc. Interspeech*, 2013, pp. 728–731.

[21] T. Hughes and K. Mierle, "Recurrent neural networks for voice activity detection," in *Proc. ICASSP*, 2013, pp. 7378–7382.

[22] G. Gelly and J.-L. Gauvain, "Optimization of RNN-based speech activity detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 646–656, 2017.

[23] S. Thomas, S. Ganapathy, G. Saon, and H. Soltau, "Analyzing convolutional neural networks for speech activity detection in mismatched acoustic conditions," in *Proc. ICASSP*, 2014, pp. 2519–2523.

[24] W. Wang, X. Qin, and M. Li, "Cross-channel attention-based target speaker voice activity detection: Experimental results for M2MeT challenge," in *Proc. Interspeech*, 2022.

[25] S.-Y. Chang, B. Li, and G. Simko, "A unified end-pointer using multitask and multidomain training," in *Proc. ASRU*, 2019, pp. 100–106.

[26] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, "Towards fast and accurate streaming end-to-end asr," in *Proc. ICASSP*. IEEE, 2020, pp. 6069–6073.

[27] T. Yoshimura, T. Hayashi, K. Takeda, and S. Watanabe, "End-to-end automatic speech recognition integrated with ctc-based voice activity detection," in *Proc. ICASSP*, 2020, pp. 6999–7003.

[28] Y. Fujita, T. Wang, S. Watanabe, and M. Omachi, "Toward streaming ASR with non-autoregressive insertion-based model," in *Proc. Interspeech*, 2021, pp. 3740–3744.

[29] Y. Sudo, S. Muhammad, K. Nakadai, J. Shi, and S. Watanabe, "Streaming Automatic Speech Recognition with Re-blocking Processing Based on Integrated Voice Activity Detection," in *Proc. Interspeech*, 2022, pp. 4641–4645.

[30] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[31] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[32] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Proc. SSPR*, 2003.

人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-061-11

任意の混合音を入力とした マイクロホンアレイ形状のキャリブレーション

Calibration of microphone array shape with arbitrary sound mixtures as input

糸山 克寿 [1,2*]　中臺 一博 [1]
Katsutoshi Itoyama[1,2]　Kazuhiro Nakadai[1]

[1] 東京工業大学
[1] Tokyo Institute of Technology
[2] （株）ホンダ・リサーチ・インスティチュート・ジャパン
[2] Honda Research Institute Japan Co., Ltd.

Abstract: 本稿では，マイクロホンアレイを構成する個々のマイクロホンの位置を観測した混合音からキャリブレーションするための手法について報告する．マイクロホンと音源の位置によって定まる音の伝達特性と音源スペクトルから混合音が観測される過程の確率的生成モデルに基づき，マイクロホン位置の事後確率の最大化によりキャリブレーションを実現する．マイクロホン数，マイクロホンアレイの形状，音源数などを様々に変化させながら提案手法によるキャリブレーションを行うシミュレーション実験により，提案手法が観測混合音に対してキャリブレーションが行えることを確認し，さらに提案手法の性質と限界について議論する．

1　はじめに

近年，マイクロホンアレイがロボットをはじめとする様々な機器に搭載されるようになってきた．また，音源定位や音源分離などの音響信号処理技術も開発・研究されている [1, 2, 3, 4]．特に，ロボット聴覚は，ロボットを含む実世界に配置可能なシステムの聴覚機能の開発を目指す研究者に注目されている [5]．

マイクロホンアレイの普及に伴い，その校正方法は非常に重要である．マイクロホンアレイをロボットに搭載する場合，デバイスの経年劣化，マイクロホン位置の測定誤差，ロボットの動作による環境変化などの理由により，あらかじめ最適に調整されたアレイのパラメータ値と最適値との間にミスマッチが生じる可能性があることが課題である．これは，図 1 に示すように，音源とマイクロホンアレイの間の伝達関数に誤差が生じてしまい，その結果として音源方向が正しく推定されない問題などを引き起こす．

使いやすいキャリブレーションには，主に 2 つの条件が必要である．1）環境音の中には複数の音源が収録されていることが多いので，同時に収録された音源信号でキャリブレーションを行うこと．2）任意の音源信号を用いて校正を行うため，特定の音を用意することなく校正が可能であること．しかし，これまでの研究の多くは，マイクロホンアレイ処理の登場以前に，手間のかかる校正方法を報告している [6, 7, 8, 9]．上記の条件を実現するためには，同時に収録された任意の音源信号で校正できる方法が必要であるが，まだ実現されていない．

そこで，本稿では，伝達関数のミスマッチ問題を回避するために，各マイクロホンの初期位置を中心とした事前分布を仮定した Maximum *A Posteriori*（MAP）推定に基づくマイクロホンアレイのマイクロホン位置の新規キャリブレーション方法を提案する．MAP 推定は，環境音としてのホワイトノイズを含む複数の同時音源を用いた校正や，環境音のような立ち上がりの悪い音源信号を用いた校正が可能である．

2　関連研究

マイクロホンアレイの伝達関数推定やマイクロホン位置推定は，非同期分散マイクロホンアレイやアドホックマイクロホンアレイなどの分野で取り組まれてきた．Thrun は，音のオンセットタイミングを利用したオンライン校正法 [10] を提案し，実際のマイクロホンデバイスを用いてその有効性を実証した．しかし，音源位

*連絡先：東京工業大学 工学院 システム制御系
〒 152-8552 東京都目黒区大岡山 2-12-1
E-mail: itoyama@ra.sc.e.titech.ac.jp

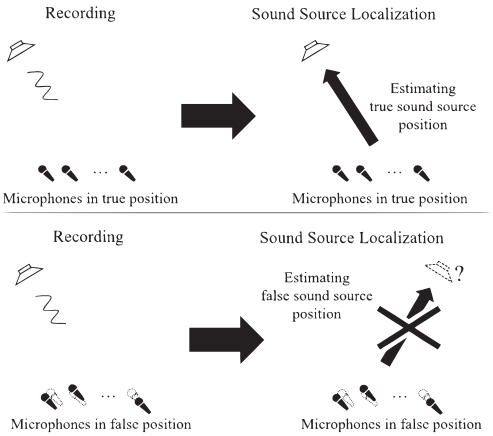


図 1: マイクロホンアレイを構成するマイクロホンの正しい位置が与えられると，それに基づく音源方向の推定結果は正しいものとなる（上段）．一方で，正しくないマイクロホンの位置が与えられると，それに基づく音源方向の推定結果も誤ったものになってしまう．

置があらかじめ決まっていること，マイクロホンが完全に同期していることなどの制約があった．三浦らは，Simultaneous Localization And Mapping (SLAM) に基づく非同期型オンラインマイク位置推定法を提案した [7]．SLAM におけるロボットの位置と地図を，それぞれ音源位置とマイクロホン位置に置き換え，さらに移動中に手拍子をすることで，8 個のマイクアレイのマイク位置を漸進的に推定することに成功した．また，アドホックマイクロアレイでは，到着時間差 (Time Difference of Arrival, TDOA) を用いた距離推定に基づく方法が提案されている [8, 9]．これらの手法では，音源信号が十分にスパースであることや，TSP (time stretched pulse) のような特定の音源信号を前提としていたため，任意の音響信号を用いられるわけではなく，その点で実用性に課題を残していた．

これらの研究では，音声信号が拍手音のようにスパースである，あるいはオンセットが明瞭であるという強い仮定を採用している．本論文ではこれらの仮定を回避するため，音源スペクトルのモデルと音源からマイクロホンまでの音の伝達過程を組み合わせた混合音スペクトルの観測モデルを構築し，そのモデルに基づいた最適化アルゴリズムを提案する．これにより，任意の音源信号の混合音を入力としたキャリブレーション手法を実現する．

3 問題設定

残響や背景雑音のない D 次元空間 (主に $D = 2$ を想定するが，$D = 3$ の状況でもほとんど同様に考えることができる) に M 個の同期されたマイクロホンと N 個の音源が存在するとする．各マイクロホンに $1, \ldots, m$ の番号を割り当て，m 番目のマイクロホンの位置をデカルト座標系で $\mathbf{u}_m \in \mathbb{R}^D$ で表す．これらのマイクロホンは，基準位置 $\bar{\mathbf{u}}_m$ にしたがって設置されているが，実際の位置 $\mathbf{u}_m$ は基準位置からのずれを含んでいる．これは，会議室などで各テーブルに一つずつマイクロホンが置かれており，実際のマイクロホンの位置はテーブル上の任意の位置であるため正確な位置は分からない，というような状況に相当する．これらのマイクロホンの位置を観測された混合音を用いて校正する（推定する）ことが本研究の目的となる．マイクロホンの位置は時不変であるとする．なお，状況を簡単にするため，1 番目のマイクロホンは座標系で原点にあるとする．

マイクロホンと同様に，各音源に $1, \ldots, N$ の番号を割り当てる．各音源はマイクロホン群から十分に遠い距離にある (far-field condition) とするため，マイクロホン群からみた方向のみを考え，音源の方向 $\mathbf{v}_n$ はマイクロホン群の中心点の近傍である座標系原点からみた音源の方向を表す大きさが 1 の D 次元の単位ベクトルで表す．$\mathbf{v}_n = (v_{m1}, \ldots, v_{mD})^\mathsf{T}$, $|\mathbf{v}_n| = 1$ 音源に関して以下の仮定をおく．

1. 音源の数は既知である．
2. 各音源の方向は既知であり，音源は移動しない．

3.1 混合音の観測モデル

音源 n からみたマイクロホン 1 とマイクロホン m の距離の差 d_{nm} は

$$d_{nm} = -\left(u_{m1}v_{n1} + \ldots + u_{mD}v_{nD}\right) \tag{1}$$

で表される．$\mathbf{u}_0 = (0, \ldots, 0)^\mathsf{T}$ なので，$d_{n0} = 0$ であることに留意する．音速を c とすると，音源 n から発せられた音がマイクロホン 1 とマイクロホン m に到達するまでの時間差は d_{nm}/c となる．音源 n から発せられた音が周波数 ω の正弦波であったとすると，2 つのマイクロホン間での位相差は $2\pi d_{nm}\omega/c$ となる．マイクロホンのサンプリング周波数を f_s，STFT フレーム長を F_0，周波数ビン数を $F = F_0/2 + 1$，周波数インデックスを $f = 0, \ldots, F$ とすると，f 番目の周波数 $\omega_f = f \cdot f_s/F_0$ となり，周波数 ω_f における位相差は $\psi_f d_{nm} = 2\pi d_{nm}\omega_f/c$ となり，周波数領域での伝達関数ベクトル $\mathbf{a}_{nf} \in \mathbb{C}^M$ は以下となる．

$$\mathbf{a}_{nf} = (\exp(i\psi_f d_{n1}), \ldots, \exp(i\psi_f d_{nM})) \tag{2}$$

n 番目の音源から発せられた音響信号のスペクトル s_{nft} が各マイクロホンで収録されたものは，

$$\mathbf{a}_{nf}s_{nft} \tag{3}$$

で表される．環境中には N 個の音源が存在するので，各マイクロホンでは全ての音源からの信号が足し合わされたものが観測される．

$$\mathbf{x}_{ft} = \sum_n \mathbf{a}_{nf} s_{nft} \tag{4}$$

ここで行列表記を導入し，

$$\mathbf{X}_f = \left(x_{mft}\right), \mathbf{S}_f = \left(s_{nft}\right), \mathbf{A}_f = \left(a_{nfm}\right) \tag{5}$$

とすると，式 (4) は

$$\mathbf{X}_f = \mathbf{A}_f \mathbf{S}_f \tag{6}$$

と書き表すことができる．

4　キャリブレーション手法

前節で導出した観測スペクトル $\mathbf{X}_f$ に基づいて各マイクロホンの位置を推定する手法について本節では述べる．音源スペクトルが確率的な生成モデルに従うと考えて，その確率モデルを定義し，観測スペクトルが従う分布を導出する．観測スペクトルに対するマイクロホン位置の尤度を導出し，この尤度を最大化することでマイクロホン位置を推定するアルゴリズムについて述べる．

4.1　確率モデル

音源スペクトルのモデルとして，複素ガウス分布を考える．

$$s_{nft} \sim \mathcal{N}_{\mathbb{C}}(0, \sigma_s^2) \tag{7}$$

$$p\left(s_{nft}\right) = \frac{1}{\sqrt{\pi}} \exp(-s_{nft}^* s_{nft}) \tag{8}$$

音源スペクトル s_{nft} がマイクロホン m で観測されたときのスペクトルは $a_{nfm} s_{nft}$ となるので，このスペクトルが従う分布は

$$a_{nfm} s_{nft} \sim \mathcal{N}_{\mathbb{C}}\left(0, |a_{nfm}|^2 \sigma_s^2\right) \tag{9}$$

となり，全ての音源スペクトルが足し合わされた観測スペクトル x_{mft} が従う分布は

$$x_{mft} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sum_n |a_{nfm}|^2 \sigma_s^2\right) \tag{10}$$

となる．すべてのマイクロホンの観測スペクトルをベクトル表現すると

$$\mathbf{x}_{ft} \sim \mathcal{N}_{\mathbb{C}}\left(0, \sum_n \mathbf{a}_{nf}^* \mathbf{a}_{nf} \sigma_s^2\right) \tag{11}$$

となる．

観測スペクトル $\mathbf{x}_{ft}$ は音源スペクトルの線形結合で表現できる．ここで，観測スペクトルに対するステアリングベクトルなどのパラメータの対数尤度を音源スペクトルの推定値に対する対数尤度で近似する．

$$\begin{aligned}\log p\left(\ldots, \mathbf{X}_f, \ldots\right) &\approx \log p\left(\ldots, \hat{\mathbf{S}}_f, \ldots\right) \\ &= \sum_{nft} \log p\left(\hat{s}_{nft}\right)\end{aligned} \tag{12}$$

$$\begin{aligned}&\log p\left(\hat{s}_{nft}\right) \\ &\overset{\mathrm{c}}{=} \Re\left(x_{m_1 ft}^* x_{m_2 ft} \exp\left(i\psi_f(d_{nm_1} - d_{nm_2})\right)\right)\end{aligned} \tag{13}$$

この対数尤度をマイクロホン位置 $\mathbf{u}_m$ に関して最大化することで，与えられた観測スペクトルに対して最も尤もらしいマイクロホンの配置を推定する，すなわち，マイクロホン位置のキャリブレーションを行うことができる．

4.2　最適化アルゴリズム

式 (12) の目的関数を勾配法を用いて最大化することでマイクロホン位置のキャリブレーションを行う．提案する最適化アルゴリズムは以下のように表すことができる．

1. マイクロホン位置の初期値を定める．
2. 前ステップの目的関数の値を $-\infty$ にセットする．
3. 反復ステップ数 t を 0 にセットする．
4. t が事前に定めた最大数 T に達するまで以下を繰り返す．
5. マイクロホン位置の現在値を用いて目的関数の値と勾配を計算する．
6. 目的関数の値の差分を計算し，その絶対値が事前に定めた閾値よりも小さければ反復を停止する．
7. マイクロホン位置を勾配と学習率を用いて更新する．
8. $t \leftarrow t+1$ としてステップ 4 に戻る．
9. マイクロホン位置を推定結果として返す．

5　評価実験

本節では，提案するキャリブレーション手法の定量的な性質およびキャリブレーション性能の限界を明らかにし，その有効性を示すために行った評価実験について述べる．シミュレーションで構築された 2 次元の無残響チェンバーに M 個の同期されたマイクロホンと N 個の音源を配置した．全ての音源から同時にそれぞれ異なる信号を発生させ，それら全てが混合した音を各マイクロホンで録音した．録音された混合音に対して提案手法を適用し，マイクロホンの位置をキャリブレーションした．キャリブレーション性能は，マイクロホンの推定位置と真の位置との平均誤差で評価する．

基本的な実験設定は以下の通りである．

- マイクロホンの基準形状：半径 10 cm の円形 8ch アレイ
- マイクロホンの実際の位置の基準位置からのずれ：平均 0 cm，標準偏差 1 cm の circulary symmetric 正規分布にしたがってサンプルされた乱数
- 音源の数：4
- 音源の配置（方位角）：0° から 360° の一様分布からサンプルされた乱数，各音源の間隔は少なくとも 20°
- 音源信号：CHiME3 challenge 孤立発話音声（7138 発話，平均長 7.64 s）からランダムに選択

以下の 3 つの実験を行った．

1. マイクロホン数または音源数を変化させた場合の性能を評価する．
   (a) 音源数を 4 に固定してマイクロホン数を 2, 4, 6, 8, 12, 16 に変化
   (b) マイクロホン数を 8 に固定して音源数を 2, 4, 6, 8, 10, 12 に変化
   (c) マイクロホン数を 16 に固定して音源数を 2, 4, 8, 12, 16, 17 に変化
2. マイクロホンアレイの基本形状を変化させた場合の性能を評価する．円形 9ch，格子状 9ch，十字形 9ch，直線形 9ch の 4 通り．
3. マイクロホンアレイの配置スケールを変化させる．半径 1 cm, 3 cm, 10 cm, 30 cm, 1 m, 3 m, 10 m の 7 通り．
4. マイクロホンの配置ずれ（真値と所与の値の差）を変化させた場合の性能を評価する．0.1 cm, 0.16 cm, 0.25 cm, 0.4 cm, 0.63 cm, 1 cm, 1.6 cm, 2.5 cm, 4 cm, 6.3 cm, 10 cm の 11 通り．
5. 与えられた音源方向に誤差が含まれる場合の性能を評価する．誤差なしの場合，標準偏差が 1°, 2°, 5°, 10° の正規分布に従う加法的誤差が含まれる場合，刻み幅が 1°, 2°, 5°, and 10° の離散値への丸め誤差が含まれる場合．

5.1　結果と考察

実験 1 の結果を Figures 2, 3, 4 に示す．これらの図は，横軸がマイクロホン数もしくは音源数を，縦軸がマイクロホン位置の平均誤差を意味し，箱ひげ図は各条件で 100 回ずつ実験を試行した結果を表している．図 2 ではマイクロホン数 M が 4 以下のとき，図 3 では音源数 N が 8 以上のとき，図 4 では音源数 N が 16 以上のとき，それぞれ推定誤差が大きく増大していることが分かる．これらをまとめると，マイクロホン数 M と音源数 N の関係が $M \leq N$ である場合に，推定誤差が大きく劣化していることが分かる．すなわち，提案するキャリブレーション手法が有効に機能するためには，音源数 N を超えるマイクロホン数 M が必要であることが分かる．ただし，ここでの音源数 N は，同時に発音している音源の数であり，例えば各音源の発話区間が与えられている場合には，マイクロホン数を超える音源数にも対応できる可能性はある．

さらに，Figure 2 を詳細に観察すると，マイクロホン数 M が 5 の場合は，マイクロホン数 M が 6 以上の場合に比べて推定誤差がわずかに大きい傾向が見られる．Figure 3 を観察すると，音源数 N が 2 または 7 の場合は，音源数 N が 3 から 6 の場合に比べて推定誤差がわずかに大きい傾向がみられる．同様に，Figure 4 を観察すると，音源数 N が 2, 14, 15 の場合は，N が 3 から 13 の場合に比べて推定誤差が大きい傾向がみられる．これらを総合すると，提案手法が有効に機能するのは，音源数 N が 3 以上で，マイクロホン数に対しておよそ 90%未満である場合であると結論づけることができる．

実験 2 の結果を Figure 5 に示す．アレイの半径が 30 cm のときが最も推定誤差が小さく，半径がそれより大きく，もしくは小さくなるほど推定誤差が増大している．特に，半径が 3 cm 以下，もしくは 300 cm 以上の場合は誤差が非常に大きくなっている．この結果から，提案手法はアレイの半径が 6 cm から 200 cm のときに有効に機能するといえる．

半径が 3 cm 以下の場合に推定誤差が増大する原因について考察する．本実験では，マイクロホン位置のずれは，平均が 0 cm，標準偏差が 1 cm の正規分布からランダムにサンプルされている．一方で，半径が 3 cm のとき，隣接するマイクロホン同士の間隔は 1.85 cm であり，これに標準偏差が 1 cm のランダムなずれを足し

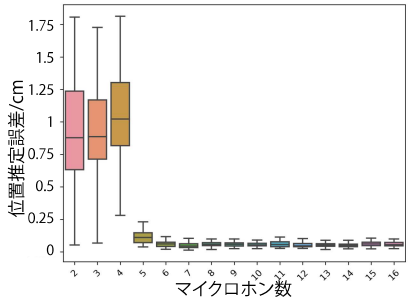


図 2: 実験 1 の結果．音源数 N を 4 に固定しマイクロホン数 M を変化させた場合

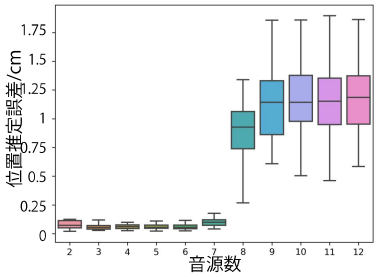


図 3: 実験 1 の結果．マイクロホン数 M を 8 に固定し音源数 N を変化させた場合

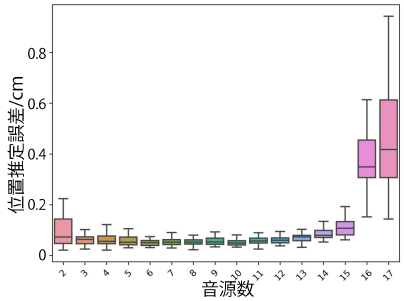


図 4: 実験 1 の結果．マイクロホン数 M を 16 に固定し音源数 N を変化させた場合

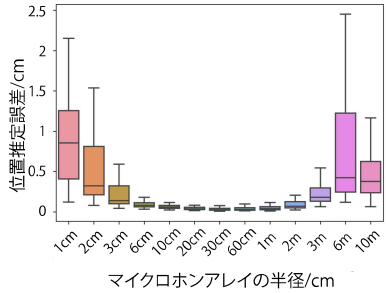


図 5: 実験 2 の結果．マイクロホンアレイの大きさを変化させた場合

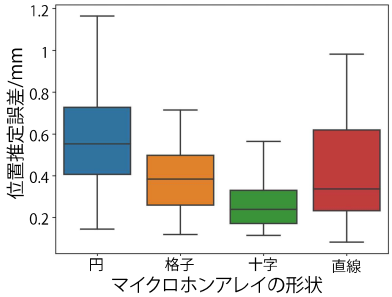


図 6: 実験 3 の結果．マイクロホンアレイの形状を変化させた場合

合わせると，マイクロホンアレイの真の形状は元の形状から大きくゆがんでしまう．校正すべき位置のずれが相対的に大きくなったことで，真の位置を推定することが難しくなり，推定誤差が増大したと考えられる．

半径が 300 cm 以上の場合に推定誤差が増大する原因について考察する．提案手法では短時間フーリエ変換で得られるスペクトログラムを入力として用いる．マイクロホンのサンプリング周波数は 16 kHz，STFT の窓長は 512 点（32 ms に相当）である．したがって，音速が 340 m s^{-1} のとき，32 ms の信号は 10.88 m の距離に渡って存在することになる．一方で，アレイの半径が 300 cm であるため，最も遠いマイクロホン同士の間隔は 6 m となり，このようなマイクロホンは同一時間フレームの信号のうち半分以下の長さしか共有していない．実際に同じ時間区間の音源信号を参照しているのは同一時間フレームの信号のうち半分以下となってしまう．このため，同一フレームであっても大部分は異なる音源信号の区間が観測である，という状況であり，推定誤差が増大したと考えられる．

実験 3 の結果を Figure 6 に示す．円形アレイが最も推定誤差が大きく，十字形アレイが最も推定誤差が小さくなった．円形 9ch アレイの場合が最も推定誤差が大きい原因について考察する．円形アレイと格子状アレイを比べると，X 軸方向・Y 軸方向におけるアレイの全長そのものはどちらも 20cm であるが，格子状アレイではマイクロホン同士の最小間隔が 10cm であるのに対して，円形アレイではマイクロホン同士の最小間隔が 6.84cm であり，すなわち円形アレイはより高密度で小規模であるとみなせる．実験 2 の結果より，半径 30cm 以下のアレイでは，スケールが小さいほど推定誤差が増大しているため，円形アレイの相対的なスケールの小ささが推定誤差の増大に繋がったと考えられる．

格子状アレイと十字形アレイは，いずれもマイクロホン同士の最小間隔は 10cm だが，十字形アレイの方が推定誤差が小さい．十字形アレイの方が端から端までの長さが大きいため，この全長の違いが推定誤差の違いに繋がったと考えられる．

直線形アレイは，80cm の端から端までの長さをもち，これはこれらのアレイの中では最大であるものの，推定誤差は最小ではない．このアレイの場合はマイクロホンが並んでいる X 軸方向における誤差は小さいものの，直交する Y 軸方向の誤差は大きく，Y 軸方向の校正はほとんど行えていない．アレイの非等方的な形状がこの結果を導き出したと考えられる．

6 まとめ

本論文では，マイクロホンアレイで観測された混合音を用いてマイクロホン位置を校正する方法を提案した．シミュレーション実験により，提案手法は任意かつ同時に複数の音源信号が録音された混合音を用いて校正できることが示され，その特性が分析された．一方，提案手法の限界として，マイクロホン数が音源数より少ない場合，アレイサイズが極端に小さい場合，または極端に大きい場合に推定誤差が大きくなることが示された．今後はこれらの限界を克服するために提案手法を改良し，さらに騒音や残響のある実環境での実験を通して，その実用性を評価することを目指す．

謝辞

本研究は科研費 JP19K12017, JP19KK0260 および JP20H00475 の助成を受けた．

参考文献

[1] C. Zhang, D. Florencio, D. E. Ba, and Z. Zhang. Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings. *IEEE Trans. Multimedia*, Vol. 10, No. 3, pp. 538–548, 2008.

[2] T. Nishiura, T. Yamada, S. Nakamura, and K. Shikano. Localization of multiple sound sources based on a CSP analysis with a microphone array. In *ICASSP 2000*, Vol. 2, pp. 1053–1056, 2000.

[3] Keisuke Nakamura, Kazuhiro Nakadai, Futoshi Asano, Yuji Hasegawa, and Hiroshi Tsujino. Intelligent sound source localization for dynamic environments. In *IROS 2009*, pp. 664–669, 2009.

[4] Kazuhiro Nakadai, Hirofumi Nakajima, Yuji Hasegawa, and Hiroshi Tsujino. Sound source separation of moving speakers for robot audition. In *ICASSP 2009*, pp. 3685–3688, 2009.

[5] Kazuhiro Nakadai, Hiroshi G. Okuno, and Takeshi Mizumoto. Development, deployment and applications of robot audition open source software HARK. *J. Robot. Mechatron.*, Vol. 29, No. 1, pp. 16–25, 2017.

[6] D. Su, T. Vidal-Calleja, and J. V. Miro. Simultaneous asynchronous microphone array calibration and sound source localisation. In *IROS 2015*, pp. 5561–5567, 2015.

[7] H. Miura, T. Yoshida, K. Nakamura, and K. Nakadai. SLAM-based online calibration of asynchronous microphone array for robot audition. In *IROS 2011*, pp. 524–529, 2011.

[8] V. C. Raykar, I. V. Kozintsev, and R. Lienhart. Position calibration of microphones and loudspeakers in distributed computing platforms. *IEEE Trans. Speech and Audio Process.*, Vol. 13, No. 1, pp. 70–83, 2005.

[9] N. Ono, K. Shibata, and H. Kameoka. Self-localization and channel synchronization of smartphone arrays using sound emissions. In *APSIPA ASC 2016*, pp. 1–5, 2016.

[10] Sebastian Thrun. Affine structure from sound. In *NIPS'05*, p. 1353 – 1360, 2005.